

# 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, Dr Mark S D Read and Prof Alessandro Mottura



## Abstract

Austenitic stainless steels have been used since the early days of nuclear power and they will play an important role in the construction of Generation III+ (Gen III+) and Generation IV (Gen IV) plant designs. Irradiation of components by neutrons is expensive with relatively few high flux reactors available. The experiments to represent 30-40 years within a reactor core are time consuming and produce radioactive waste as a by-product. Two alternatives to testing by irradiating with a neutron source are either irradiating with a proton beam or by modelling damage events with a computer.

The efficiency and computational power of computers continues to improve. First principles Density Functional Theory (DFT) and molecular dynamics (MD) are computational methods that are now able to give insights into why materials behave the way they do.

1. Proton sources are more widespread, they are relatively cheap and may be focused into a much smaller beam to concentrate the amount of damage per time unit. The radioactive waste may be significant but by carefully selecting the ion energy the electromagnetic repulsion between ions and nuclei reduces transmutation of target atoms whilst maintaining the amount and depth of damage into the material required for testing purposes.

A modified Bateman equation is derived to calculate the radioactivity of a proton irradiated target and a computer program (Activity) is created to implement the calculation using evaluated nuclear reaction cross section data. The program is compared to data measured from the irradiation of an iron sample using the University of Birmingham cyclotron. It is also used to model the radioactivity of a target irradiated to 100 displacements per atom (DPA) at a range of beam energies in order to minimise both the irradiation time and radioactive waste. The results show an increase in target activity of 3 orders of magnitude for a 25MeV proton beam when compared to a 10MeV proton beam.

2. Sensitisation of austenitic steels, by processes such as welding, depletes Cr from grain boundaries removing the corrosion resistant  $\text{Cr}_2\text{O}_3$  passive layer. Previous (experimental) work has shown that these steels, when doped with Pd or Ru, retain corrosion resistance at the grain boundary[1]. This work takes a step towards investigating whether or not these platinum group metal (PGM)s deplete or saturate at the grain boundary while under irradiation and this may be studied with MD simulations. A computer program is developed to fit interatomic potentials to experimental data and DFT generated data. The properties of face centered cubic (FCC) Fe are computed using DFT. These data along with other properties and DFT generated data are used to derive interatomic potentials (with a dominant ZBL pair for close range collision interactions) for Fe-Pd. These potentials are a step towards MD simulations of the irradiated grain boundaries.

# Contents

# List of Figures

# List of Tables

# Chapter 1

## Introduction

*Several Gen III+ nuclear reactors have been proposed for construction in the UK (with Hinckley Point C now under construction) and Gen IV nuclear reactors are 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 inter granular stress corrosion cracking (IGSCC) can be addressed. Doping these steels with platinum group metal (PGM)s has been seen to reduce IGSCC, but the effects on corrosion resistance are unknown for these steels when irradiated by a radiation field.*

### 1.1 Motivation

I have often questioned the motivation behind this work, and at times it has been a challenge to see the wood through the trees. To help me focus throughout I would restate the motivation and objectives to myself.

Mass produced steels are not perfect repeating crystals, but are made up of small grains. Cr is added to steel to make stainless steel, and this is more resistant to corroding than steel. When steel is in a nuclear reactor, it will have to perform under extreme conditions, such as:

- high temperatures
- strain cause by high pressures and radiation induced swelling
- radiation damage and strains resulting from this damage
- a changing environment, for example radiolysis of water and transmutation of isotope due to radiation
- corrosive environments while in the reactor
- corrosive environments out of the reactor (e.g. fuel cladding in storage)

Radiation damages the steel in a number of ways, including directly knocking atoms out of their place within the lattice structure of the crystalline grains that make up the steel as well as changing the isotopes that make up the steel into new isotopes. An example of the latter is a neutron reacting with an Fe atom, transmuting it into a Co atom.

In time, the radiation damage causes the percentage of Cr at the surface of the grains to drop, and as it falls the steel loses its protection from corrosion at the boundary between the grains it is made of.

This work is divided into two parts.

### 1.1.1 Part 1: Activity Computer Program

The materials must be tested before being used in a nuclear reactor. One way to do this would be to place samples of the steel into a test reactor. This is expensive and, as a by-product, the steel sample becomes radioactive. It is difficult to create a large number of neutrons, but it is much easier, and cheaper, to create a beam of protons. Protons can be accelerated in a machine such as the Cyclotron at the University of Birmingham.

The damage that protons cause to a sample is not precisely the same as that caused by neutrons, but it is a cheaper alternative and is a trade-off between the cost and results. One side effect that proton irradiation shares with neutron irradiation is the creation of radioactive waste.

The first part of this work investigates exactly how radioactive a sample becomes when irradiated with a proton beam. An existing set of equations, named after Mathematician Harry Bateman, were modified, and a computer program was created to perform the calculation. The user inputs the constituent elements that make up the material, the ratio of these elements, and the irradiation settings. The program then estimates how radioactive the sample will be and the predicted gamma energies.

### 1.1.2 Part 2: Iron-Palladium and Iron-Ruthenium Potential

The addition of Cr to make stainless steel is not the only way to make a steel resistant to corrosion. Adding metals such as Mo, GlsPd and Ru to steel can increase the resistance to corrosion, but Mo is several hundred times the cost of Fe ore, and Pd is thousands of times as expensive.

Simulating radiation damage using a computer is now a feasible and sensible way to investigate how these materials will be affected by radiation damage, and the simulations may reveal insights that experiments are not able to show, either because they happen on too small a timescale or within the material at too small a length scale.

Key to the simulations success is being able to accurately calculate how the atoms interact with one another. The second part of this work concentrates on deriving a mathematical description of how Pd & Fe and Ru & Fe atoms interact with one another, which would then allow future simulations of steel with small amounts of Pd or Ru added to it.

Radiation causes Cr to be depleted at the grain boundary. If these simulations show that Pd and Ru are not depleted, it would suggest that the corrosion resistance is maintained, despite the decrease in the amount of Cr at the grain boundary due to radiation damage.

## 1.2 Nuclear Power in the UK

### 1.2.1 Nuclear Fuel

The primary fuel of commercial nuclear power stations since those first built in the 1950s has been U. U-235 is fissile and fissions with a high probability when thermal neutrons are captured, due to its very high thermal neutron fission cross section (fig. 1.1).

Natural U contains a small percentage of fissile U-235 (0.7%) with the rest being its heavier isotope, U-238. U-238 is fissionable, but its fission cross section is much lower than U-235 (fig. 1.1). Many of the design considerations for reactors in operation revolve around the amount of U-235 there is in the fuel and having a sufficient flux of thermal neutrons passing through the fuel to sustain a reaction.

A number of Gen IV designs use the fast neutron spectrum to fission U-238 and other fuels such as Th-232 to generate energy and breed other fissile isotopes. By moving away from U-235, this increases the potential fuel stockpile, but also brings its own set of challenges.

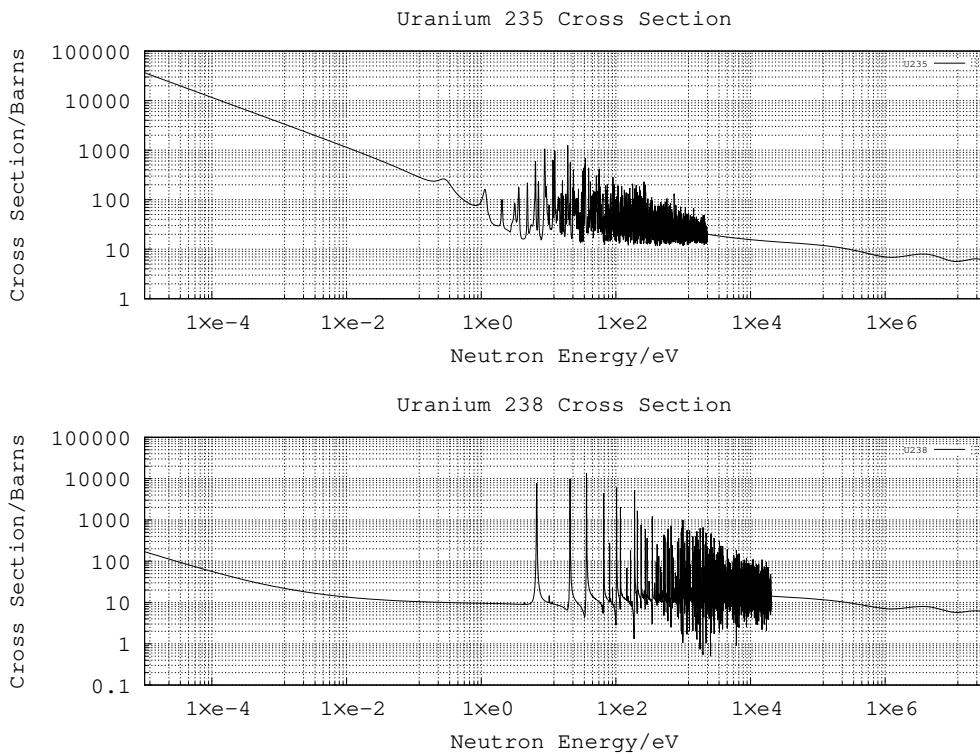


Figure 1.1: U-235 and U-238 Fission Cross Sections

### 1.2.2 Generation I Reactors

The first generation reactors were primarily prototype reactors. They included graphite moderated reactors, light water and heavy water reactors. Early reactors were designed to produce electricity, but also fissile material for nuclear weapons. With a power output of less than 2MW, the Gen I power station Calder Hall generated much less electricity than a modern nuclear power plant which may produce 1GW per reactor. It did, however, produce Pu for the UK's nuclear weapon programme.

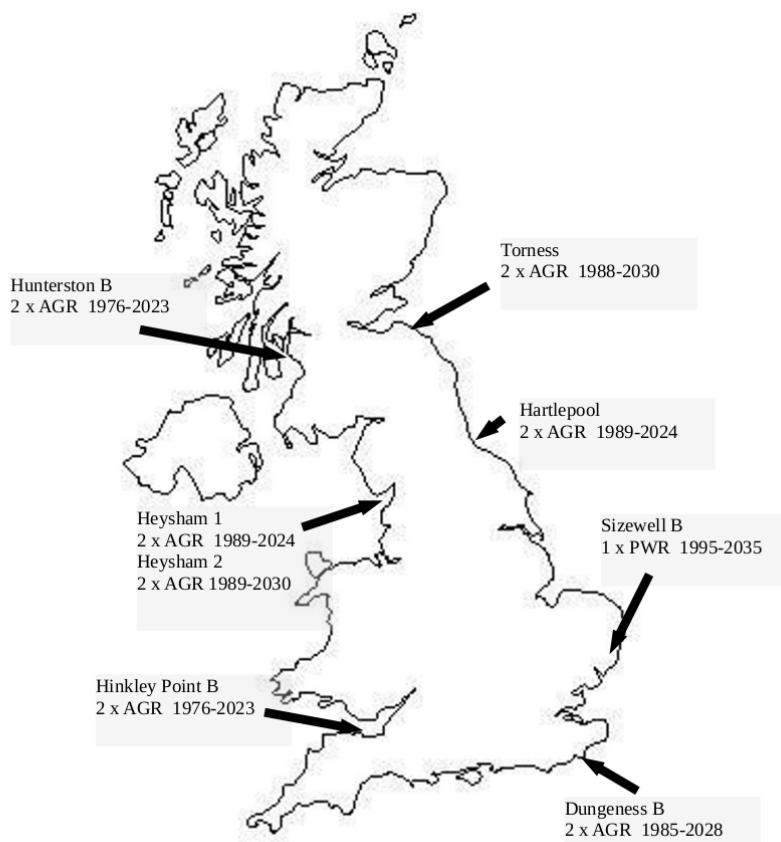
Magnox type reactors were the first used in the United Kingdom. These reactors used natural U as a fuel and were carefully designed to produce energy despite using an un-enriched fuel. Graphite was used as a moderator, and the low neutron capture cross section of the Magnox Mn alloy cladding allowed a nuclear reaction to occur despite the fuel not being enriched.

In all there were 11 Magnox power stations built in the UK with 26 reactors in total. Construction of the first commercial reactor in the UK, Calder Hall, started in 1953 and it operated from 1956 to 2003. All have now shut down with the last, Wylfa in Anglesey, closing in 2015.

### 1.2.3 Generation II Reactors

Gen II marked the transition from prototype reactors to higher powered commercial reactors. There were a number of designs including the AGR, a graphite core  $CO_2$  cooled reactor, that was implemented in the UK. Light water reactors included the PWR and BWR in the west and the VVER in the USSR. The RBMK was a graphite moderated reactor used in the USSR and the CANDU a heavy water reactor designed and used in Canada.

There are many of these reactors from this period still in operation around the world, and there have been recent implementations of these designs. There are 15 reactors currently operating in the UK, and they are all Gen II reactors. Of these, 14 are AGR and 1, Sizewell B, is a PWR (fig. 1.2).



*Figure 1.2: Remaining reactor locations in the UK*

The planned life span for this generation of reactor was 30-40 years, although this has now been extended to 50-60 years in some cases. The AGR reactors in the UK have been operational for over 40 years, and by the time Hunterston B and Hinkley Point B are planned to be decommissioned, they will have been operational for 46 years. Designed with safety improvements in mind over Gen I reactors, they were also intended to produce more power with less of a focus on producing fissile material for weapons. Unfortunately, this generation of reactors had the worst track record of any with safety, with Three Mile Island, Fukushima and Chernobyl all being Gen II designs.

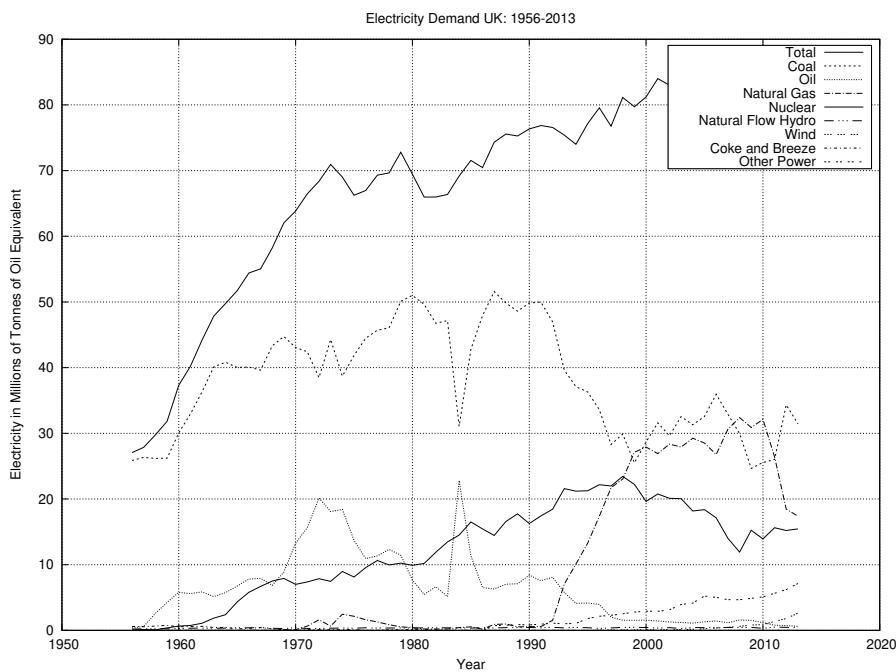
Gen II+ reactors have been built recently with 18 CPR-1000s being constructed over the last decade in China. The UK is currently looking towards Gen III+ reactors while Gen IV proposals are being researched.

#### 1.2.4 Generation III Reactors

There are no Gen III reactors operating in the UK. The first was built in Japan in 1996 and the type of reactor installed was an Advanced Boiling Water Reactor (ABWR). Bradwell B is a proposed site for the Gen III Hualong One type PWR power plant (HPR1000), and if this goes ahead it would have an expected opening date between 2030-2035.

This generation of reactor was designed to exceed the service life of the previous generation, and generate power for at least 60 years. The safety features have been improved and include increased resilience to external factors, including accidental or intentional aircraft collision into the plant[2].

Gen III+ reactors go further in adding passive safety features, such as natural circulation and gravity coolant feed [3]. Other aims of future generations are to standardise and simplify designs whilst improving the economy over Gen II+ and Gen III.



*Figure 1.3: Electricity in Millions of Tonnes of Oil Equivalent*

## 1.3 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 (fig 1.3). 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[4], and the remaining PWR reactor, Sizewell B, is expected to remain operational until 2035[4].

There is an obvious concern that within the next ten years the UK will lose a sizeable proportion of its electricity generation capabilities. There are increasing pressures on countries to reduce their carbon dioxide output, so the gap created by ageing nuclear plants and coal power needs to be filled.

### 1.3.1 Why choose Nuclear Power?

As a civilization we need energy, whether it's in the form of electricity or stored chemically, and whilst we are becoming more efficient at using that energy, the demand for it will remain (unless something drastically changes our society). There is a choice between burning fossil fuels or bio fuels, using energy from the Sun, wind, ground, oceans or rivers and finally using the energy of the nucleus, whether by fission or fusion.

Each has its drawbacks and advantages. Renewable energies are unreliable; wind power only provides energy when there is a wind, and if the wind is too strong, they must be shut down or risk damage. The Sun is only available for a portion of the day, and the duration and intensity change with the seasons, not to mention the impact of clouds on solar power. Renewable sources are not very good at responding to demand; there isn't a button to turn up the power of the Sun, or increase the velocity of the wind when the national grid demands it. If we were completely reliant on renewable energies, there would either need to be an efficient way to store energy on a large scale, or many more solar, wind, tidal and hydroelectric power stations than would be needed to produce excess energy. Whilst the energy source may be free, turbines, solar panels, dams and so on require maintenance, so an excess of these would be wasteful and costly.

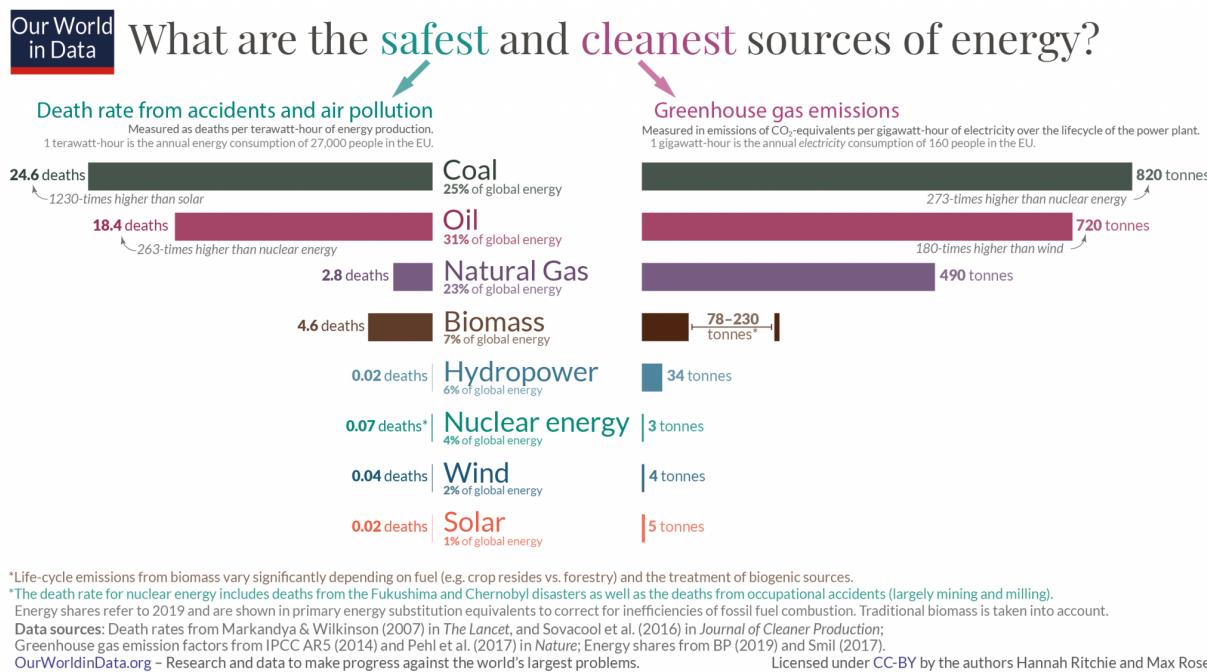


Figure 1.4: Safest and cleanest sources of energy [7] - deaths per terawatt hour and tonnes of CO<sub>2</sub> per gigawatt hour

Fossil fuels release carbon dioxide, sulphur dioxide, naturally occurring trace radioactive elements and other pollutants into the atmosphere. These power plants are useful because the fuel is cheap, but the waste is put back into the environment without much processing and the energy produce may be varied to meet the demand of the national grid.

Nuclear power has a bad reputation with the public, in part caused by accidents such as the Windscale fire, Three Mile Island, Chernobyl, and Fukushima. There have been examples of poor handling of nuclear waste in the past, such as the legacy storage ponds at Sellafield, and long term storage in geologically stable areas is something that hasn't been achieved to store the existing waste, let alone waste created in the future.

Modern nuclear power plants are very expensive to build, with the proposed 3.2GWe Sizewell C power plant expected to cost £20 billion or more [5]. When compared to £0.5 billion for the 0.884GWe Carrington CCGT Power Station [6], the initial costs are £0.57 billion per GWe for CCGT compared to £6.25 billion per GWe for Nuclear.

There is an effort around the world of countries aiming to reduce the production of carbon dioxide as a result of burning fossil fuels. When a power plant is operational, the carbon dioxide produced by nuclear power is negligible when compared with gas, oil and coal power plants.

Despite its reputation amongst the public, nuclear is much safer than fossil fuels and is on par with renewable energies. The pollution caused by fossil fuels affects us all and is responsible for many deaths each year, causing more misery than nuclear power ever has (fig. 1.4). Coal and oil are responsible for 350 and 260 times as many deaths as nuclear.

As a species, we have used combustion of chemicals to generate heat for thousands of years, culminating in high efficiency CCGT power stations, with efficiencies above 60%. We are at the beginning of our journey with nuclear power, with many new and innovative designs proposed, where safety to plant workers and the general public are a priority.

Future designs may help to tackle the problematic waste generated in the past, and by using fertile material to produce fissionable elements other than U<sup>235</sup>, the amount of fuel available to reactors will increase.

### 1.3.2 Planned and Under Construction in the UK

Several sites have been acquired with the aim of building new nuclear power stations in the UK. There are five sites and three reactor designs[8]:

- Hinkley Point: two Areva EPRs (EDF Energy), Hinkley Point C under construction
- 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) ...

Two Gen III+ reactors are under construction in the UK: Hinkley Point C 1 and C 2. These reactors are Areva NP designed Gen III+ EPRs, which are PWRs, and the proposed opening years are 2025 and 2026 respectively.

There are also plans to construct a third plant at Sizewell that, if accepted, will be an EPR reactor. Sizewell C will join the decommissioned Sizewell A and operational Sizewell B, that may operate until 2055.

### 1.3.3 Proposed Generation III+ Nuclear Power Plant Designs

#### Areva EPR

The 1.6GW Areva EPR design has four primary loops transferring heat by pressurised water from the reactor to heat exchangers. It requires U enriched to 5% 235U in the form of U Oxide Pellets. The inlet temperature is 568.75K and the outlet temperature is 602.95K.

There are 241 fuels assemblies, each containing 265 fuel rods giving a total of 63865 rods. The fuel rod cladding is made from 316 stainless steel and has an inner diameter of 7.72mm and an outer diameter of 9.68mm. The materials used to construct the control rod drive mechanisms includes 410 stainless steel and 304 stainless steel.

There are two EPR reactors planned for the proposed Sizewell C power plant.

#### Westinghouse AP1000

The 1.1GW Westinghouse AP1000 design is a pressurised water reactor with two primary loops transferring heat from the reactors to heat exchangers. An enriched  $UO_2$  fuel, up to 5% 235U, is clad in Zirlo (a proprietary Zr alloy).

Zirlo is an alternative cladding material to 316 stainless steel, and as a cladding material is absorbs fewer neutrons than steel cladding.

Property	316SS Fe 20Cr 8Ni 1Mo	Zirlo (Hill Approximation)
Bulk Modulus (GPa)	164.9 [9]	98.5 [10]
Shear Modulus (GPa)	74.6 [9]	33.2 [10]
Young's Modulus (GPa)	194.3 [9]	89.7 [10]
Poisson's Ratio	0.30 [9]	0.35 [10]

Table 1.1: Bulk, shear, Young's modulus comparison: Zirlo and austenitic stainless steel

The inlet temperature is 552.55K and the outlet temperature is 597.85K which is close to the temperature range of the EPR. The composition of the control rod absorber material includes 304 stainless steel.

### 1.3.4 Generation IV

#### Goals of Generation IV Reactors

The GenIV International Forum has put forward four main goals for this next generation of nuclear power[11]:

- sustainability
- safety and reliability
- economics
- proliferation resistance and physical protection

The selection of known materials, and the development of new materials, will play a key part in all four of these goals.

#### Carnot's Theorem

Whatever the motivation, whether it is to increase profits or to supply energy at greater amounts and for less cost to the consumer, increasing the efficiency of usefully using energy is critical. Solar panels are continually being improved to edge closer to their theoretical limit and wind turbines are being constructed larger and with more advanced materials to extract as much energy from the wind as possible.

In the same way, power plants that use heat are constantly being developed to improve efficiency. In the nineteenth century Carnot showed that the maximum possible efficiency of a heat engine is determined by the difference in temperature between the heat reservoirs.[12]

$$\eta_{max} = 1 - \frac{T_c}{T_h} \quad (1.1)$$

To increase maximum efficiency, the temperature difference should be increased, and this leads to higher temperature reactors. There will be a trade off between the increased temperature, the ability of the materials to withstand the temperature, health and safety considerations, the lifetime of components, the effect on the coolant and more.

The first generation of reactors in the UK were the gas cooled Magnox reactors. With core temperatures of approximately  $620K$ [13], the thermodynamic efficiency was relatively poor. This was limited by the  $MgO$  cladding, which was in turn selected due to the fuel. Combined cycle gas turbines have much higher temperatures within their turbines, and the temperature of the steam within the secondary steam turbine can reach  $850K$ [14], significantly higher than in a Magnox reactor.

The second generation of power plants, particularly in the UK, included the advanced gas reactor that used  $C_2$  as a coolant. This reactor design increased temperatures to over  $870K$ , and thus the maximum possible thermodynamic efficiency was increased.

Several Gen IV reactors have designed operating temperatures that exceed those of the AGR, and this brings a new set of challenges to overcome.

## Fast Reactors

Examples of thermal neutron reactors include Magnox, AGR, LWR and Candu reactors. These reactors contain moderators, designed to slow neutrons, decreasing their energy to thermal temperatures and leveraging the large neutron fission cross section of U-235. The cross section for fast neutrons and U-235 is in the region of 1 barn, but thermal neutrons and U-235 have a fission cross section of over 1,000 barns.

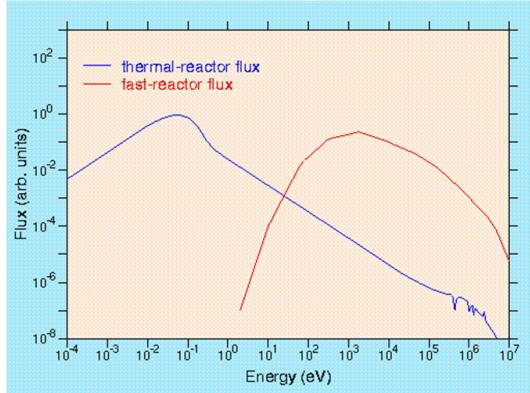


Figure 1.5: Typical Reactor Flux - Thermal and Fast Spectra

Natural U contains 0.7% U-235 and 99.3% U-238[15]. The thermal fission cross section for U-238 is small for thermal neutrons, and can be measured in the microbarn to millibarn range, but it has a much higher fission cross section with fast neutrons with a cross section closer to 1 barn for neutrons with an energy of 1MeV or more. The neutron flux in a fast reactor is spread over a higher energy band than that of a thermal neutron reactor (fig. 1.5).

Fast neutron reactors have been tested and used since the 1950s, and there are several Gen IV reactor designs based on this approach. Given the much larger percentage of U238 in natural U, Fast Neutron Reactors have a larger stockpile of fuel. They also breed fissile isotopes, Pu 239 and 241. This has its advantages and disadvantages. The advantage is clear, as it produces fissile fuel as it runs, but the creation of fissile materials is a concern as these isotopes could be used in nuclear weapons.

## Lead-cooled Fast Reactors

Light elements such as the hydrogen in water molecules, or carbon in a graphite stack, allow neutrons to efficiently lose energy. When neutrons scatter inelastically with heavier elements, such as lead, they lose a much smaller percentage of their energy.

The Lead-Cooled Fast Reactor (LFR) uses lead as the coolant. Lead has a low reaction cross section and, because of its mass relative to a neutron, the kinetic energy lost in collisions is low, keeping more neutrons in the fast energy group. As the coolant is a liquid, and the boiling point of lead is over 1,970K, the system will run at atmospheric pressures and any problems associated with void formation due to boiling is removed. The molten lead will act as a gamma shield and it will be chemically less reactive than the coolant in a Sodium-Cooled Fast Reactor (SFR) or Supercritical-Water-Cooled Reactor (SCWR).

There are several disadvantages to using lead as a coolant. The melting point of lead is 600K, so the reactor would need to be heated first for the coolant to become a liquid. This also restricts the lowest temperature and thus the maximum efficiency possible that may be extracted from the system without the coolant forming a solid. The density of lead also poses a problem for the structure needed to support the reactor.

European Lead Fast Reactor (ELSY) is a 600MWe Lead-Bismuth eutectic cooled fast neutron reactor. It has a lower melting point than lead, but there are concerns for the transmutation of Bismuth to Polonium. The 9,000

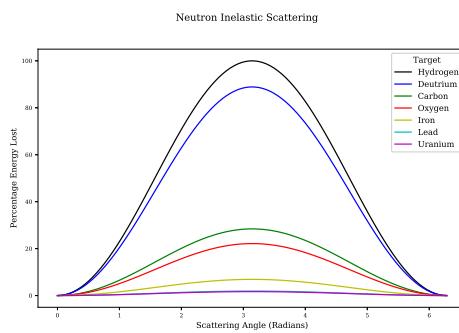


Figure 1.6: Inelastic scattering - scattering angle vs energy loss for a range of target atoms

tonne molten lead coolant and the high flux of fast neutrons[16] will be challenging conditions to overcome.

## VHTR

Traditional nuclear power plants, as well as coal and the secondary cycle of CCGT plants, boil water to drive turbines. Very-High-Temperature Reactor (VHTR) designs use thermal neutrons and a fissile fuels. Designs plan to have outlet temperatures of up to 1,270K[17]. With a helium coolant, there are several options.

- directly drive turbines
- heat water to create steam and drive turbines
- use the high temperature to help create hydrogen, and combine with steam driven turbine

Hydrogen may be extracted from high temperature water either by a thermo-chemical, high temperature electrolysis or hybrid process[18], and they all benefit from the very high temperatures provided by the predicted outlet temperatures (fig 1.7).

The Pebble-Bed Reactor (PBR) is a type of VHTR where spherical fuel pellets are held in a hopper shaped reactor. Temperatures within the core may reach 1,770K, and this will pose a challenge for materials scientists.

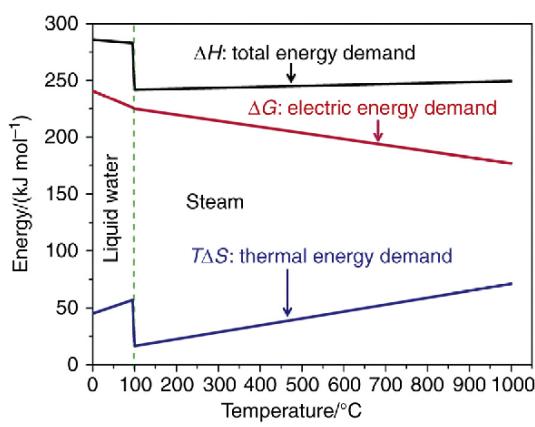


Figure 1.7: Energy demand to produce hydrogen by electrolysis as temperature changes

## GFR

The Gas-Cooled Fast Reactor (GFR) design is similar to the VHTR, but rather than use thermal neutrons, it will use fast neutrons. The design will use a ceramic core and have outlet temperatures up to  $1120K$ . This will give similar options to the VHTR where high temperatures would be used to assist in the production of hydrogen, or it would be used to power a turbine with a secondary steam powered circuit similar to that of a CCGT.

There are challenges ahead, including the development and testing of advanced materials that can withstand these temperatures while under high energy neutron flux. However, unlike LFRs, SFRs and SCWRs, the coolant is chemically inert and isn't transmuted by neutrons removing the risk of the coolant becoming radioactive.

## SFR

Memories of small amounts of sodium, kept under oil in school chemistry labs, being carefully cut and prepared ahead of a violent reaction with water, probably come to mind at the first mention of an Sodium-Cooled Fast Reactor (SFR). It may not be the first choice of an element to use as a coolant, but it does have advantages.

Sodium has a low melting point, of just under  $370K$ , and a boiling point of  $1,150$ . With an operating temperature of up to  $820K$ , the reactor would be able to run at atmospheric pressure and without the issues associated with the coolant boiling. Its density is  $0.971gcm^{-3}$ . Comparing this to the LFR the coolant would be 12 times less dense and this would have advantages when designing the structure.

There have been several SFRs built and operated using this technology, such as the Russian BN-600 reactor. This particular reactor has a sodium primary loop which heats a secondary sodium loop which in turn heats a third loop for water and steam.

## SCWR

At  $647K$  and a pressure of  $22.1MPa$  (approximately 218 atmospheres), water becomes supercritical (fig. 1.8) which is neither a liquid or a gas[19]. The current generation PWR and BWR operate at approximately  $501 - 598K$  at  $15.2MPa$ [8] and  $551 - 560K$  at  $7.1MPa$ [20] respectively.

Supercritical water exists above  $647K$  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 Supercritical-Water-Cooled Reactor (SCWR) is the only Gen IV reactor design that uses water as the coolant[4]. The economic benefits have already been seen in SCW fossil fuel power stations, and it is incorporated in Gen IV water cooled fast and thermal reactors.

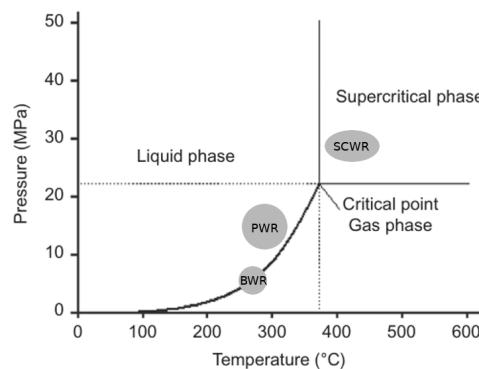


Figure 1.8: Phases of Water: Pressure vs Temperature

There are side effects to using supercritical water. It is an oxidant and is now being considered to “burn” organic waste in future waste processing plants. The combination of supercritical water chemistry and irradiation damage must be considered, as well as higher temperature and pressure, when considering the materials used by a SCWR.

### Molten Salt Reactor

MSRs have been operated for over 50 years. Chemically, salts are very stable, and this has obvious safety benefits over reactors like the SFR. The coolants are fluoride salts that have high boiling points which gives the added safety protection of being able to operate at atmospheric pressure, unlike a PWR where a pressure vessel is needed.

The fuel is dissolved into the molten salt. There are no solid fuel rods to place into the core, remove or reprocess. The molten salt is processed on the site to remove poisons, waste and add new fuel.

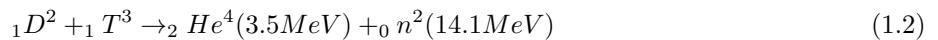
MSRs may use either thermal or fast neutrons and a range of fuels including Th. This is particularly interesting as Th is three times as abundant as U and would increase the fuel available to us to use to generate nuclear power.

A safety feature that takes advantage of the fuel being dissolved in the molten salt are large drain tanks under the reactor. A plug is designed to melt if it reaches a certain temperature and this would quickly drain the molten salt from the reactor into large and cold drainage tanks. The reactivity would drop and the fuel would cool reverting it to a solid salt.

Salts considered in various designs are chlorides, nitrates and fluorides. Corrosion of the reactor due to the molten salts is a concern. Elements that provide protection to corrosion, such as Cr, are prone to dissolution into the molten salt[21]. Where the fuel is suspended in the molten salt as Tristructural Isotropic (TRISO) fuel particles, formation of chromium-carbide precipitates may be an issue.

### 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 International Thermonuclear Experimental Reactor (ITER) 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 (eq. 1.2). 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.

Unfortunately, nuclear fusion as a commercial method of creating energy, is often said to be 30 years away, and this has been the case for some time now. It is a lot to ask, recreating a process that occurs within a star, in a safe and steady manner. Once this problem is solved, it could mark a turning point for human civilization.

## 1.4 Damage to Nuclear Core Components by Neutrons

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 excite electrons, promoting them to higher energy levels, or ionise the atoms where there is enough energy to eject electrons completely from atoms. They 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. With enough energy, neutrons cause large damage cascades within the material.

One fuel source for many of the Gen III+ and Gen IV reactors is U-235, whether as enriched U or otherwise. The neutron(s) released by the fission of U-235 atoms have a spectra of energy (fig. 1.9) 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).

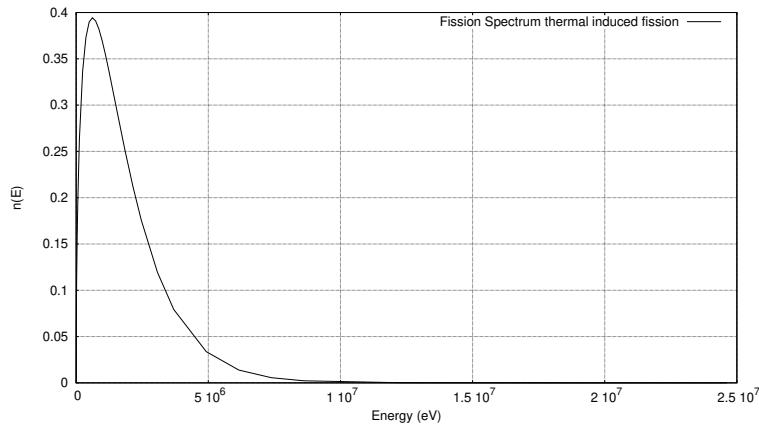


Figure 1.9: Neutron Energy Spectrum from Fission[22]

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 have enough energy to create a great deal of damage, whereby they transfer kinetic energy to an atom in the material that become the primary knock-on atom (PKA) in a cascade of atoms through the material. Neutrons can travel deep into a material, creating many damage cascades through the material until they lose enough energy or pass through the material completely.

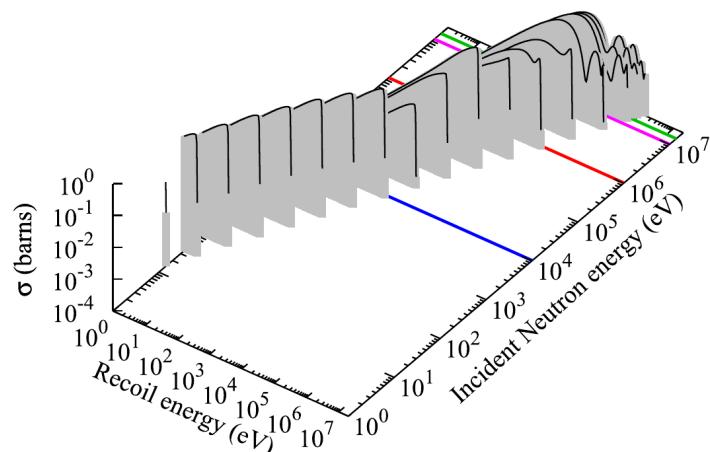


Figure 1.10: PKA energy as a function of neutron energy in Fe[23]

The intensity of neutron energies from the fission of U235 peaks around 1MeV (fig. 1.9). The energy transferred

to PKAs will vary depending on the mass of the target nucleus and scattering angle but, in Fe at least, 1MeV neutrons will create the majority of its Fe PKAs with energies ranging from 1keV to 100keV (fig. 1.10).

## 1.5 Radiation Damage: Replacing Neutrons with Protons

It is difficult to generate large fluxes of neutrons. When we pick up a cup, press keys on a keyboard, they react because of the electromagnetic force. Neutrons have no net charge, so they cannot be controlled in a similar way. Nuclear reactors are the most common way to create large fluxes of neutrons.

This is an expensive method, and one that may be inconvenient as the sample being tested would need to be placed inside a reactor. Rather than do this, ions may be used to replicate the damage caused by neutrons, but in a more controlled and accessible way.

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 electromagnetic 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.

Thanks to detailed reaction cross section data files and decay data, the amount of radioactivity induced by both protons and neutrons can be calculated. This will assist in the selection of proton beam parameters to replicate irradiation damage, to the required DPA, whilst keeping the radioactivity at a safe level.

## 1.6 Simulation Material Damage

Irradiating and testing materials irradiated by both neutrons and protons have a number of drawbacks, including expensive facilities and the creation of radioactive waste. Simulations may be used to help understand the processes on a mesoscopic scale and make predictions on how the material in question may behave inside a reactor under certain specified conditions.

Part of this work focuses on the damage of stainless steel by neutrons. The damage process occurs on time scales and sizes too small to capture experimentally and play back to study. The theories may not be sufficiently well understood, or are often unsolvable with current techniques. Modelling helps bridge the gap between theory and experiment.

### 1.6.1 Molecular Dynamics

Molecular dynamics models have been used to simulate radiation damage. Damage cascades caused by primary knock-on atom (PKA)s with energies of 10, 20 and 50KeV have been modelled in BCC W[24], whilst cascades caused by PKAs of 5, 10 and 20KeV have been simulated in BCC Fe[25].

Neutrons have a long mean path before interacting with a target when compared to ions. It would be impossible with current and foreseeable technology to simulate a neutron travelling through a material because the volume of the material, and the number of atoms and interactions between atoms, would be so large. Rather than attempt this, it is more productive to focus on individual damage cascades caused by PKAs in order to conserve computing resources.

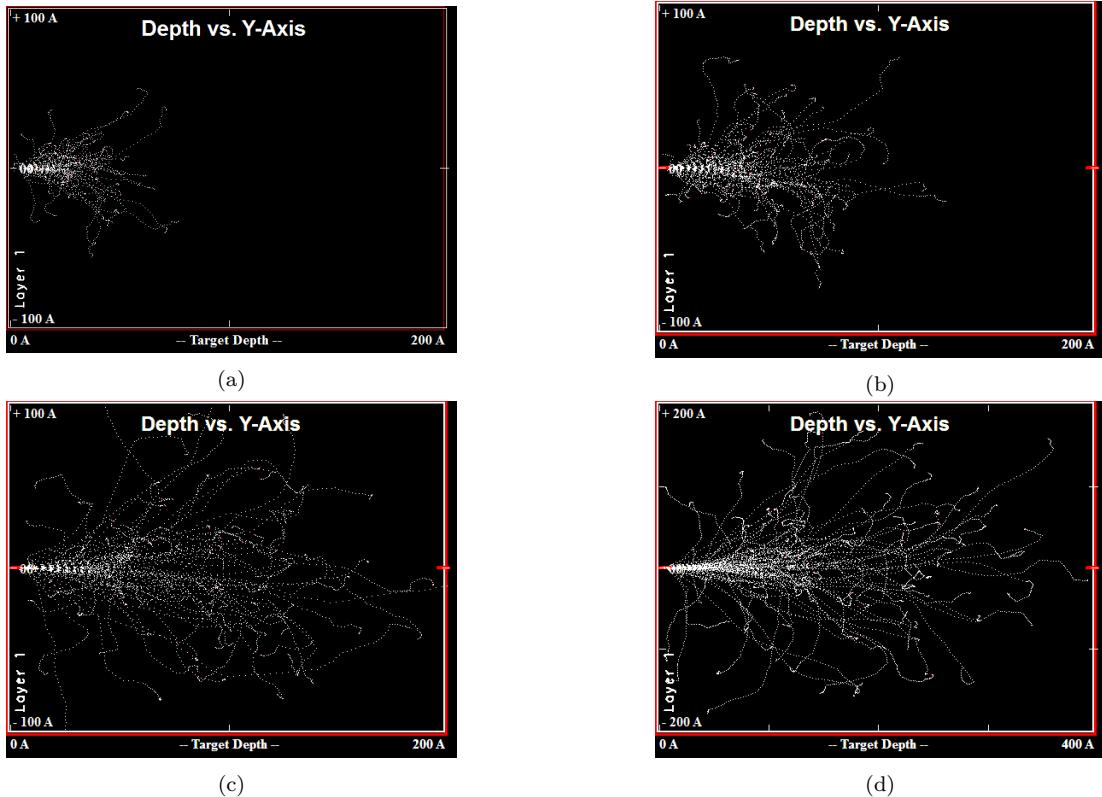


Figure 1.12: (a) 5KeV Fe PKA Damage Cascade in Fe (b) 10KeV Fe PKA Damage Cascade in Fe (c) 20KeV Fe PKA Damage Cascade in Fe (d) 50KeV Fe PKA Damage Cascade in Fe

Molecular Dynamics simulations require interatomic potentials to govern how the atoms in the simulation interact with one another. Typical simulation sizes range from thousands of atoms to millions of atoms.

Fe has a density of approximately  $8.5 \times 10^{-2}$  atoms per cubic angstrom, and this leads to simulation sizes between almost one million atoms and fifty million atoms for cascades with PKAs ranging from 5KeV to 50KeV (fig 1.12, table 1.2).

PKA Energy (KeV)	Volume ( $\text{Ang}^3$ )	Atoms
5	$1.0 \times 10^6$	$9.0 \times 10^5$
10	$6.0 \times 10^6$	$5.4 \times 10^6$
20	$8.0 \times 10^6$	$6.9 \times 10^6$
50	$6.4 \times 10^7$	$5.5 \times 10^7$

Table 1.2: Approximate simulation sizes by PKA energy

The interatomic potentials used to simulate a damage cascade must be able to model the very close separations during the collisions event. For very close separations, an exponential potential such as the Ziegler-Biersack-Littmark (ZBL) potential may be used, and for other separations a standard potential such as the embedded-atom method (EAM) may be used, in particular for metals.

There have been attempts to create potentials based on Physics, but the nuances are still poorly understood and there aren't any models that can be applied accurately to any material type under any set of circumstances. A number of recent interatomic potentials, including a selection of transition metals, have been fit to match the experimental data rather than the underlying physics.

## 1.6.2 Density Functional Theory

In quantum mechanics, the Schrödinger equation and the wavefunction of a system describes all that can be known about that quantum system. Unfortunately, it is very difficult to compute when there are just several electrons in the system. Many atoms in a system makes this an impossible task with current technology.

Density Functional Theory (DFT) replaces the many electron wavefunction with a single electron wavefunction. Rather than use the positions of all the electrons, which would result in  $3^n$  parameters, the charge density, which varies in space with just 3 parameters, is used.

DFT allows for the calculation of the forces between atoms and the energy of that collection of atoms by replacing the many parameters to give the coordinates of many electrons with the three coordinates for the charge density. From the forces the optimum (relaxed) basis vector and atom positions may be computed, and from the energy of multiple configurations other properties are computable (bulk modulus, elastic constants, etc).

## 1.6.3 Interatomic Potentials for Molecular Dynamics

Interatomic potentials help to describe the interaction between atoms. Originally they were used for pairs of atoms and a collection of atoms on a pair by pair basis, but in the 1980s the complexity of potentials increased as they started to include the effects of the background electron density that the atoms are embedded in. This lead to the Finnis-Sinclair (FS) and embedded-atom method (EAM) type potentials.

Whilst DFT is a purer form of computational materials science, more closely linked to physics, it is computationally intensive and suitable for small systems of hundreds to thousands of atoms. By using interatomic potentials to recreate the forces and energies that would be computed by DFT, the systems may be increased in size to thousands to millions of atoms, and these calculations may be run over a time frame with hundreds or thousands of time steps.

There are a wide range of potentials, and these are typically selected based on the type of material and the required accuracy. Simple pair potentials may still be used, but angularly dependent potentials have been developed for materials with covalent bonds, and metals use FS and EAM potentials. The potentials are made up of a set of functions, and these functions change from derivation to derivation.

# Chapter 2

## Background: Ionizing Radiation

*There are a wide range of radiation sources and types of ionising radiation. They interact differently with matter, depending on the type of radiation and its energy. Damage may be caused with the radiation transferring energy directly to the target. The target may be transmuted or the surrounding, for example the creation of Cobalt-60 in steel in the nuclear industry. The environment may be changed chemically, and one example of this is the radiolysis of water.*

### 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, neutrons, ions and Ggammas, and two of these (neutrons and protons) are of particular interest in this work. Each of these are capable of ionizing the atoms they interact with, given sufficient energy. Higher energy and momentum radiation it is able to directly damage materials, or the DNA structure of organic life, that it interacts with. Lower momentum ionizing radiation may change the water chemistry of the environment or within the cells of organic lifeforms that is detrimental to life.

#### 2.1.2 Protons and other 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 higher energy levels in their atoms, or by removing electrons from atoms altogether, causing the ionization of those 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. Inelastic scattering is also a possibility, leaving the target nucleus in an excited state.

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.

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.

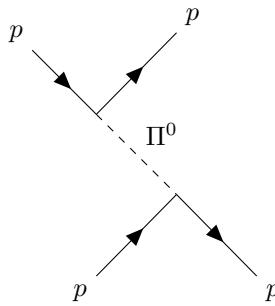


Figure 2.1: Direct interaction p-  
p[27]

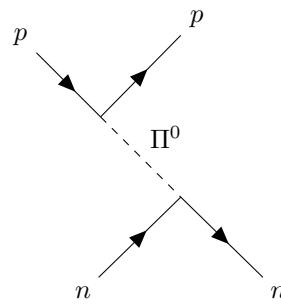


Figure 2.2: Direct interaction p-  
n[27]

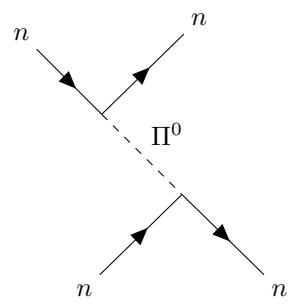


Figure 2.3: Direct interaction n-  
n[27]

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 further radiation 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 unpaired electrons by a dipole-dipole interaction. Neutrons also interact with the nucleus of atoms, and this is dependant upon the target atom and the energy of the projectile neutron. The energy of neutrons may be grouped as shown in table 2.1, and as mentioned earlier, thermal neutrons are of interest to current reactor designs, and fast neutrons are of interest to several future designs.

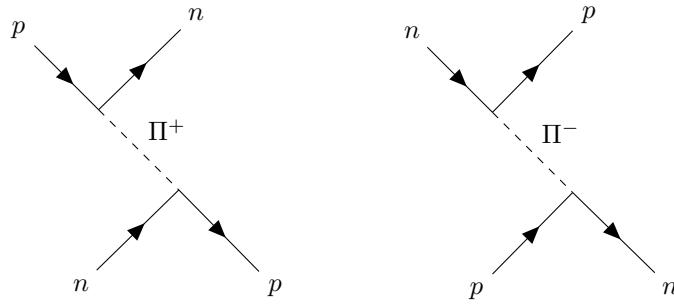
Name	Energy Range	Velocity/ms <sup>-1</sup>	Wavelength Ang
Cold	0-0.025 eV	$0.0 - 2.2 \times 10^3$	$> 1.8$
Thermal	0.025 eV	$2.2 \times 10^3$	1.8
Epithermal	0.025-0.4 eV	$2.2 \times 10^3 - 8.8 \times 10^3$	0.5-1.8
Cadmium	0.4-0.6 eV	$8.8 \times 10^3 - 1.1 \times 10^4$	0.4-0.5
Epicadmium	0.6-1.0 eV	$1.1 \times 10^4 - 1.4 \times 10^4$	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	$2.4 \times 10^5$	$2.9 \times 10^{-4} - 0.02$
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 [26]

The nuclear force holds nuclei together by exchanging pions. It is 137 times stronger than the electromagnetic force, but because of the mass of the pion (approximately 130-140MeV) it only acts over a short range of approximately 1fm. The nuclear force is attractive but has a short-range repulsive core.

For nuclear reaction and damage calculations, the interaction of neutrons with matter is represented by a cross section (section 4.4.3). Cross sections are used to calculate the probability of a certain interaction occurring and are dependent upon the target nucleus, how much of the material the pass through and the energy of the neutron.

The attenuation of neutron radiation is therefore a complicated process. A much larger thickness of shielding is required to stop neutrons, or at least bring them to thermal temperatures. Not only this, but the thermalized neutrons must then be captured. During this process various isotopes within the attenuating shielding will have

Figure 2.4: Exchange interaction n-p  
[27]Figure 2.5: Exchange interaction p-n  
[27]

been transmuted, possibly into radioactive isotopes. As a rough estimate if a certain amount of material will attenuate neutrons to 10% their original flux, then the same amount of material will reduce the flux further to 1%. This is however an over simplification and computer codes such as Monte Carlo N-Particle (MCNP) and TART more accurately model the transport of neutrons through materials.

#### 2.1.4 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 (eq. 2.1).

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

#### Pair Production

The energy of photons in the database used for 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 (fig. 2.6).

#### Compton Scattering

An incident photon with enough energy may interact with the electron of an atom, and if it transfers enough kinetic energy it will eject the electron from the atom (fig. 2.7). A lower energy photon is also created that

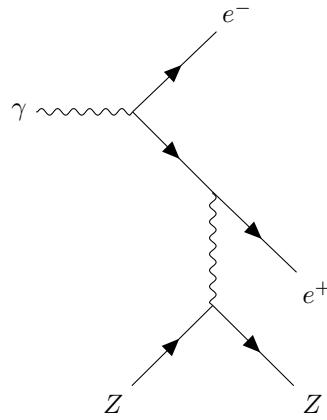


Figure 2.6: Pair production Feynman diagram

carries away the remainder of the energy, but also linear momentum, as both energy and momentum must be conserved.

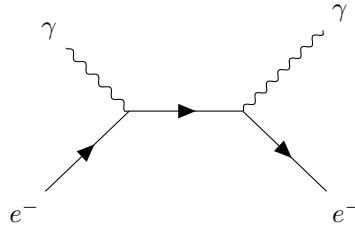


Figure 2.7: Compton Scattering Feynman diagram

### Photoelectric Effect

Photons of sufficient energy interact with and eject electrons from the surface of a metal (fig. 2.8). At lower energies, the photons have no effect on the electrons. Ejected electrons have a maximum kinetic energy equal to the difference between the energy of the interacting photon and the workfunction,  $T_{max} = h\nu - \phi$ .

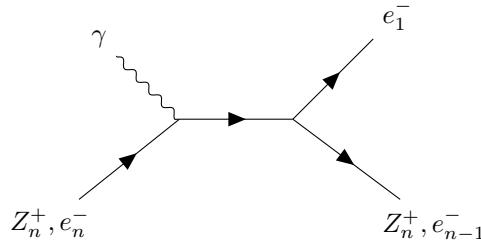


Figure 2.8: Photoelectric Effect Feynman diagram

### Coherent Scattering

Lower energy, non-ionizing photons, have insufficient energy to eject electrons (fig. 2.9). Coherent scattering is the interaction in which photons change direction, scattering, without losing energy.

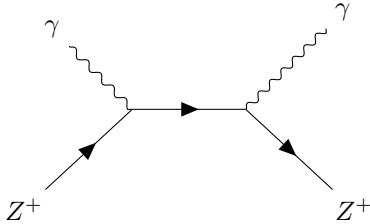


Figure 2.9: Coherent Scattering Feynman diagram

### Gamma Attenuation

Whilst an infinite amount of material is required to reduce the intensity of gammas to zero, they may be reduced by a significant amount with dense material. Lead, Uranium and other dense materials with a large number of electrons per unit volume are ideal.

$$I(\mu/\rho(E), t) = I_0 \exp\left(\frac{-\mu}{\rho}(E)\rho t\right) \quad (2.3)$$

The attenuation is dependent upon the thickness of the material , with a thicker material reducing the intensity of the gammas (eq. 2.3[28]). It is also dependent upon the mass attenuation coefficients that are in turn dependent upon the photon energy. A table of attenuation coefficients are available for each element[29].

## 2.2 Microscopic and Macroscopic Cross Sections

As discussed above there are a range of different ways in which matter interacts. The probability that a certain interaction will happen, which may be a function of the target, projectile and projectile energy, is described with the microscopic cross section. The total microscopic cross section is computed by summing all the individual reaction channels (eq. 2.6)

$$\sigma_{total} = \sigma_{elastic} + \sigma_{inelastic} + \sigma_{absorption} + \sigma_{fission} + \dots \quad (2.4)$$

The cross section has units of area and is measured in barns. When this area is multiplied by the target thickness and number density of the target, the fraction of projectiles that will react in this way is computed (see section 4.4.2).

Multiplying the microscopic cross sections by the number density defines the macroscopic cross section, whether it be for an individual reaction type or the sum of them all. This has units of atoms per meter and taking the inverse gives the mean free path of the projectile in the target material.

$$\Sigma_{total} = N_d \times \sigma_{total} \quad (2.5) \qquad \lambda = \frac{1}{\Sigma} \quad (2.6)$$

The probabilities and associated libraries of a neutron, proton, deuteron and so on transmuting a target atom are discussed in more detail in section 4.4.3.

## 2.3 Quantifying Radiation and its Effects

The reason for attributing a value to a radiation source is usually because we want to quantify it and link that value to the effects it will have on materials, equipment and, most importantly, people. There are a number ways of quantifying radiation, an associated number of different units as well as many other factors that affect how concerned we should be about a particular source.

Parameters most obvious to quantify radiation are the rate at which the particles are being created as well as the duration the equipment, material or person is exposed to that radiation.

The energy of the particles will determine the type of damage it is able to do. We are regularly exposed to visible light with several hundred watts or more per square meter courtesy of the Sun. The same power from a 1-2MeV gamma source would require far fewer photons, with a million visible light photons for every gamma, but the affect wouldn't be a gentle warming as it would be for sunlight, but a fatal dose of radiation.

How far the radiation is able to travel is another factor as a certain dose may be received very close to the source, but reduced due to both spherical degradation and attenuation at a distance further away. This is particularly relevant for charged particle radiation such as alphas and betas.

The material, component or part of the body radiation passes through will also determine its effect. Some parts of the body are more sensitive to radiation than others.

Radioactivity, the number of decays per second, is measured in either Becquerel, Curie or Rutherford. The Curie was based on the radioactivity of Radon-226, but for the most of this work, the unit of choice will be Becquerel.

Fluence ( $\phi$ ) is the measure of the total number of particles passing through an area (eq. 2.7). Whilst this is typically defined as the amount passing through a spherical area, the proton radiation that concerns this work is directional and it will be the planar fluence[30]. The fluence rate is a measure of how many particles pass through a sphere (or plane where planar) per second.

$$\phi = N/A \quad \dot{\phi} = \frac{d\phi}{dt} \quad (2.7)$$

Absorbed dose is the amount of energy absorbed by a unit mass. This will be the sum of the fluence and energies of each individual particle and will depend upon the proportion of that energy deposited in the target. It is measured in the SI unit Gray although a historical unit is Rad, which is equal to 1 centigray.

$$\begin{aligned} D_{equivalent} &= W_r \times D_{absorbed} \\ D_{equivalent} &= W_r \times W_t \times D_{absorbed} \end{aligned} \quad (2.8)$$

Both the type of radiation and the type of tissue determine how much of an impact that radiation has per unit energy per unit mass. Both the equivalent and effective (eq. 2.8) dose are measured in the Si unit Sievert, and historically they have been measured in Rontgen Equivalent Man where the base unit is cGy rather than Gy.

## 2.4 Effects of Radiation on Organics

In this work the primary types of radiation considered in this work are neutrons, protons and gammas. The materials damage will be predominantly due to neutrons, protons and cascade atoms. Damage to humans will

be primarily due to gamma rays.

It is assumed the process causing the damage is screened from humans and that this radiation will no longer be present when the irradiated targets are being handled. It is also assumed that and beta and alphas will also be screen with the main source of irradiation that poses a threat to humans being gammas. For this reason more focus will be given to the effect of gammas on humans.

### 2.4.1 Reference Levels of Radiation

Although a sievert is only one joule per kg, because of the nature of the damage cause by ionising radiation, safe levels are measured in millisieverts per year. The annual upper limit for the general public is just . A dose of just 5 Sieverts is enough to kill 50% of people within 1 month (table 2.2).

Exposure	Dose (mSv)
Dental X-ray	0.005
Chest X-ray	0.02
UK average annual dose limit for the public	1.0
UK average annual dose (background)	2.7
Whole body CT scan	10
Nuclear industry employee annual exposure limit	20
Acute radiation effects	1,000
50% lethal dose within 1 month	5,000

Table 2.2: Irradiation dose comparison in mSv

The regulations on ionising radiation[31] also specify limits per square centimetre on skin, where the total dose might be within the annual limit but when concentrated in a small region it could cause damage to that region. Certain parts of the body also have limits applied, such as the lens of the eye.

### 2.4.2 Levels of Radioactivity from Neutron Irradiated Components

In work by Rodenas and Verdu a MCNP5 simulated neutron flux environment and a thermal flux of  $2.5 \times 10^7 n/cm^2 s$  a stainless steel cylinder (20mm diameter and 70mm depth) was irradiated for 10 hours[32]. The sample was composed of 65% Fe, 19% Cr, 2.0% Mn, 1.0% Si by weight as well are small amounts of C, Ti, P and S. After 10 minutes cooling, it was calculated to have an activity of over  $1.0 \times 10^7$  Bq and, by several orders of magnitude, it was predominantly due to the creation of Mn<sup>56</sup>.

The maximum cross section of Mn<sup>55</sup> ( $n, \gamma$ ) Mn<sup>56</sup> is 2.0 barns and the estimated reaction rate at this flux is just over  $2.0 \times 10^6$ . As the half life of Mn<sup>56</sup> is short compared to the irradiation time (a half life of 2.5 hours) the creation rate of Mn<sup>56</sup> and decay rate equalise, saturating the sample with Mn<sup>56</sup>.

A simple neutron activation code was developed, which will be discussed in more detail in this work, and the predicted Mn<sup>56</sup> activity vs time was calculated (fig. 2.10). The overall gamma dose, for 1 minute of exposure to an 80kg person standing 2m from the source was also calculated (fig. 2.11). The calculated activity for Mn<sup>56</sup> was less than that predicted by Rodenas and Verdu [32], but it was within an order of magnitude. It predicts a total activity after cooling of  $2.3 \times 10^6$  Bq with almost 90% of this due to Mn<sup>56</sup>.

The flux in the within the flux trap of the HFIR is much higher, with high energy neutrons having a flux of  $1.0^{14}$  neutrons per cm squared per second (fig. 2.12). A second calculation was performed for this increased flux, also irradiated over 10 hours (fig. 2.13). The total activity increased from  $2.3 \times 10^6$  Bq to  $7.1 \times 10^{12}$  Bq and the overall dose per minute was much higher, between 40 and 50 milligrey per minute over the first hour of cooling. These are dangerous levels of radiation.

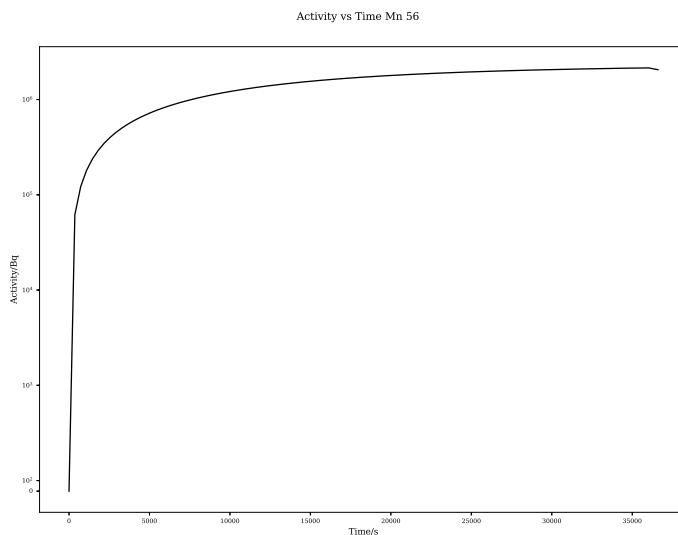


Figure 2.10: Predicted Mn 56 activity after neutron irradiation

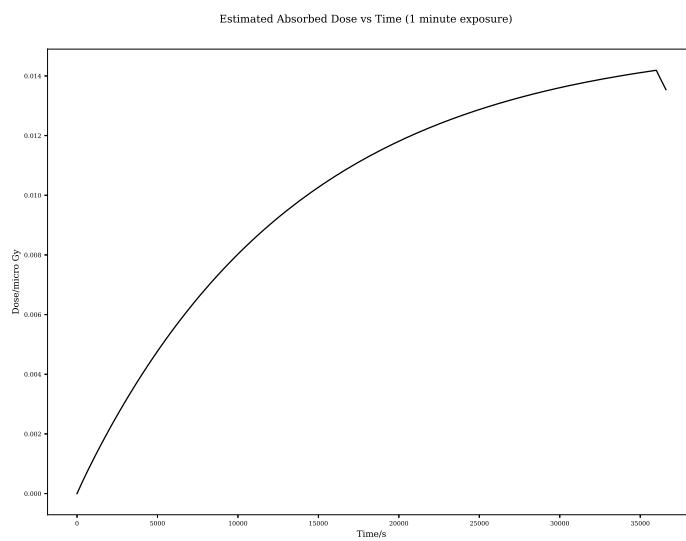


Figure 2.11: Steel Irradiation Gamma Dose  $2.5 \times 10^7$  neutrons per cm squared per second

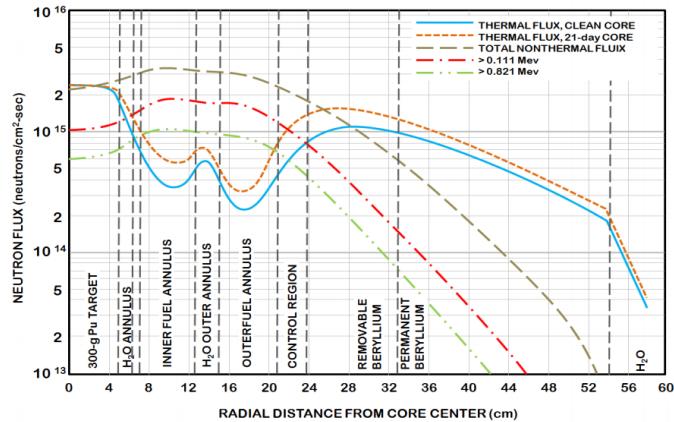
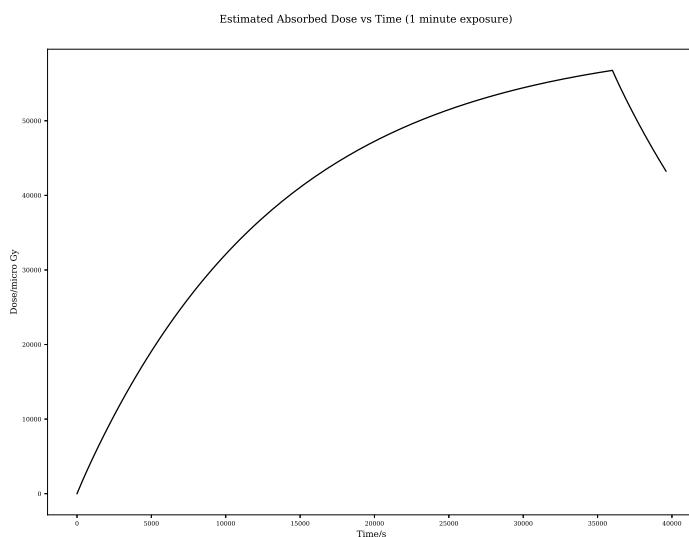


Figure 9: Neutron Flux at the MidPlane of HFIR (at 85MW)

Figure 2.12: High Flux Isotope Reactor Neutron Flux



*Figure 2.13: Steel Irradiation Gamma Dose  $1.0 \times 10^{14}$  neutrons per cm squared per second*

### Damage to Organics by Radiation

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 copied.

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.

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.

The damage caused results in either deterministic effects or stochastic effects.

Where there's a minor exposure to radiation, the damage caused to cells and DNA may lead to cancer later in life, or it may cause issues that are inherited from the irradiated person. These are stochastic effects as there's a chance they could happen, but it's not certain.

Deterministic effects are due to larger doses where cells are severely damaged. There's much less chance involved and the effects will be reproduced if multiple people are exposed. Up to a certain dose, there is still a chance that a person would survive, but the point where the chance of death is 50% is the LD50, and for radiation the dose is 5Sv (table 2.2).

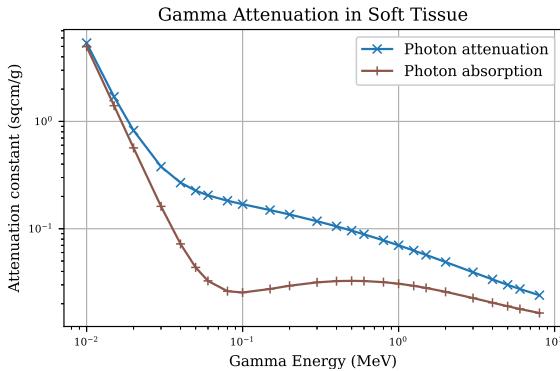


Figure 2.14: .

The body relies heavily on three type of blood cell: erythrocytes (red cells), leukocytes (white cells) and thrombocytes (platelets). These are produced from stem cells within bone marrow and in the course of a day billions are replaced in the body. Large doses of radiation destroy the cells quicker than they can be produced and this leads to death.

The lens of the eye is very sensitive to radiation and cataracts can be created by gamma rays. Radiation can also cause damage to the gut including removal of the lining, bleeding and death. These are examples of deterministic effects.

Whilst it is difficult to determine the effects of radiation, particularly for stochastic effects where the effect is heavily dependent on chance, this work on calculating activity of proton irradiated targets aims to give the user a predicted activity and dose value with associated assumptions and caveats. It is then down to the user to apply the resulting data to the situation and evaluate any health consequences.

## 2.5 Damage Cascades in Metals

When a metal is damaged by radiation, there are microscopic defects created and removed in a very short space of time. The result of many months or years of damage is complex and will be discussed in more detail in chapter 3.

The cohesive energy of atoms are measured in eV (table. 2.3), whereas the energy of incoming neutrons or fission fragments are measured in MeV. Radiation in the form of electrons, protons and heavy ions will lose energy to electrons in the target as they pass through. They may also lose energy to atoms within the lattice, knocking that atom out of place and creating a PKA. Due to its neutral charge and magnetic moment, neutrons have a very weak interaction with charged particles. They primarily lose energy directly with the nucleus of atoms within the target, also creating a PKA.

Element	Cohesive Energy/eV
Aluminium	-3.36
Iron	-4.32
Palladium	-3.91
Platinum	-5.77
Zirconium	-6.32

Table 2.3: Cohesive Energies [33]

The PKAs create a damage cascade, and their mean recoil energy is dependent upon the energy and type of radiation. After each cascade, there will be a period of time where the material quenches, with a recombination

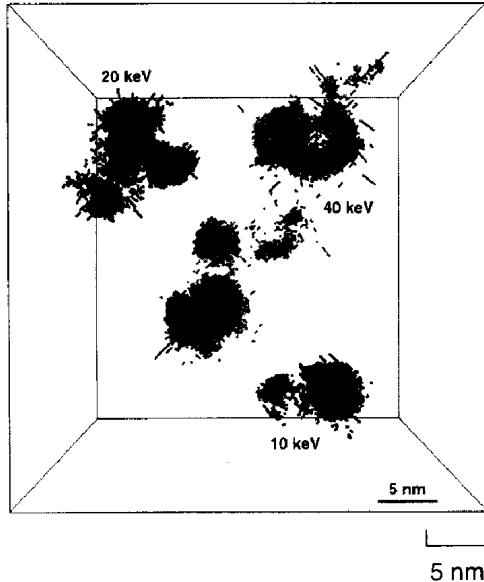


Figure 2.15: Interstitials in the cascades of 10keV, 20keV and 40keV displacement cascades [35]

of some interstitials and vacancies. The PKA creation and cascade thermal spike cover a time range of  $10^{-18}$ s to  $10^{-13}$ s, with the quenching phase lasting in the region of  $10^{-11}$ s. However, the cumulative effect of radiation damage on a component within a reactor core may only become apparent after months or years.

Projectile	Mean Recoil Energy
1 MeV Electrons	60eV
1 MeV protons	200eV
1 MeV heavy ions	5keV
1 MeV neutrons	35keV

Table 2.4: Average recoil energy of Nickel PKAs[34]

The immediate aftermath of interstitials created three damage cascades is shown in fig. ???. Damage cascades have been the subject of a number of studies and due to their short time scale and complexity, involving thousands to millions of atoms, are studied with computer simulations.

# Chapter 3

## Background: Austenitic Steels and Nuclear Power

*Austenitic stainless steels have played an important role in the nuclear industry. They have a good resistance to corrosion and behave well at moderately high temperatures, having a good resistance to creep. They have been used within the primary circuit and secondary circuit of power stations, as cladding for fuel pellets, in the steam dryers, pre-heater tubing, core structurals, primary piping and more. Ferritic steel is magnetic and has a BCC structure, whereas austenitic steel is non-magnetic with a FCC structure, with its high Nickel content stabilising this phase. An area of concern is the susceptibility of austenitic stainless steels to IGSCC.*

### 3.1 Stainless Steel

#### 3.1.1 Introduction

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% Cr in 1911. The addition of Cr to this level causes the formation of a passive protective layer of chromium oxide.

Fe is alloyed with Cr, Nickel and Carbon in varying quantities to make Stainless Steel. Depending on the application, other elements, such as Molybdenum, may be added to enhance the properties of the steel.

#### 3.1.2 Grades of Stainless Steel

While the criteria that qualifies an alloy as Stainless Steel is containing Fe, C and at least 10.5% Cr, there are many grades that have variety of properties and atomic structures. The composition influences the structure, resistance to corrosion, yield strength, whether it is magnetic or not, and more.

Both Fe and Cr are BCC at standard temperatures and pressures, but adding austenite stabilisers such as Ni, Mn or C, the structure of the steel can be altered from BCC to FCC. Balancing the proportion of these elements changes the phase of the steel, and this may be represented as a Schaeffler diagram (fig. 3.1) or a ternary phase diagram (fig. 3.2).

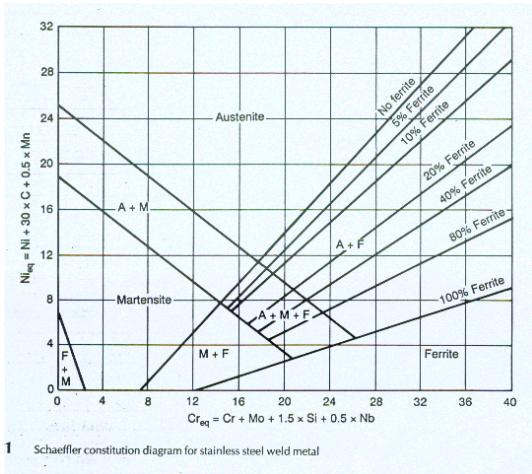


Figure 3.1: Steel Cr-Ni Schaeffler Diagram

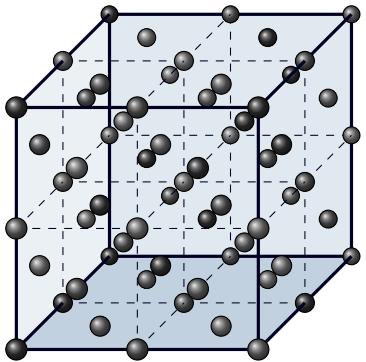
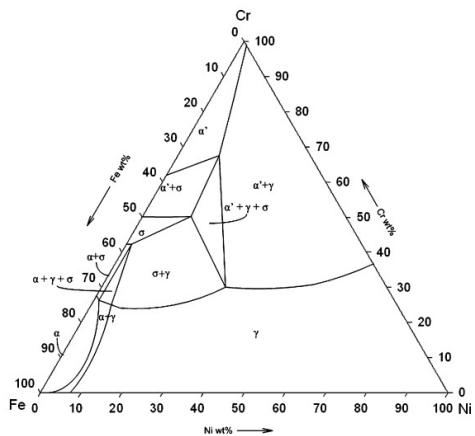


Figure 3.3: BCC structure of ferritic stainless steel

### Ferritic Stainless Steel

At room temperatures and pressures, Fe exists in the alpha phase, which is a BCC crystal. It is energetically favourable for the magnetic moments of the Fe atoms to align with one another ferromagnetically. Ferritic stainless steels also have the natural alpha phase crystal structure of pure Fe at room temperature which is BCC (fig 3.3). 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 American Society of Engineers codes 405 and 430.

### Austenitic Stainless Steel

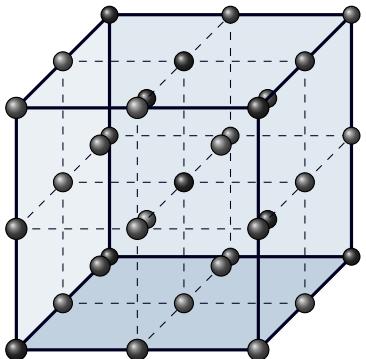


Figure 3.4: FCC structure of austenitic stainless steel

Austenite is a FCC allotrope of Fe (fig 3.4), 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 10.5 wt% or more Cr they require an austenite stabilising element such as carbon and/or nickel to be added.

Two examples of such steels are ASME codes 304 and 316. Both have a high Cr content, in the region of 18-20%, which is in excess of the minimum passive film requirement of around 10-11%. The natural structure of such an Fe Cr alloy would be BCC, however 304 and 316 Steels contain Nickel (approximately 8% and 10% respectively) which is an austenite stabiliser, and this is responsible for the FCC structure of the steel. The 316 grade contains a minimum of 2% Mo to improve its resistance to corrosion.

Ferritic and martensitic have a better resistance to thermal or swelling shock than austenitic [37], but the corrosion resistance of austenitic steels is a deciding factor when choosing a steel for an application where corrosion resistance is important.

### Martensitic Stainless Steel

Martensitic stainless steels have a FCC structure at high temperature, but when heat treated take on a BCC structure. This allows martensitic, unlike austenitic and ferritic, to be hardened by heat treatment. These steels are magnetic, they contain more than 10.5% Cr but have a much lower Ni than austenitic grades, if any. They are corrosion resistant, but the corrosion resistance of austenitic steel is typically better. Two examples of such steels are ASME codes 410 and 431. A comparison of properties between common grades of each steel are detailed in table 3.1.

Grade	Code	HRB	Proof strength (MPa)	UTS (MPa)	Density ( $kgm^3$ )	E (GPa)
Austenitic	304	80	230	540-750	7900	200
Austenitic	316	79	240	530-680	8000	200
Ferritic	405	75	250	400-600	7700	220
Ferritic	430	85	280	450-600	7700	220
Duplex	255	32	550	750-1000	7800	200
Martensitic	410	90	205	600	7700	215
Martensitic	420	95	345	700	7700	215

Table 3.1: Common grades of ferritic, austenitic, duplex and martensitic stainless steel and their properties. Hardness using the HRB, ultimate tensile strength (UTS) and Young's modulus E.

## 3.2 Austenitic Steels in Nuclear Reactors

### 3.2.1 The Use of Austenitic Steels in Reactors

Due to their excellent strength and corrosion resistance, austenitic steels have been widely used in nuclear reactors. The cheaper to manufacture 304SS and more corrosion resistant 316SS have been particularly popular choices of steel. There are drawbacks, when compared to other alloys including ferritic and martensitic steels, and two in particular are IGSCC and swelling.

### 3.2.2 Atom Damage Cascades

Fission fragments carry away the majority of the energy released during fission, with an approximate energy of 170MeV per  $^{235}\text{U}$  fission event. The fragments are large and have a highly charged positive nucleus that stops rapidly due to the Coulomb interaction with the nuclei of surrounding atoms. Electrons released by beta decay travel a greater distance; they lose energy by interacting with electrons, but may also knock atoms out of place. Neutrons travel further still and lose energy by interacting directly with the nuclei of the surrounding material.

When an incoming projectile hits an atom (the primary knock-on atom (PKA)), it is knocked out of its place and loses energy by colliding with other atoms and to the electrons in the metal. Depending on the energy of the PKA, this causes a damage cascade. Atoms knocked out of their regular position in the crystal lattice leave vacancies behind. The displaced atoms either recombine with vacancies, become interstitial atoms or diffuse to defect sinks. A number of complex problems arise as a result of this damage, and this can take many years and high damage doses to appear.

### 3.2.3 Damage Rates

The safety of a Nuclear reactor is the primary concern; it must generate power, but it must be safe. It must also be cost efficient. Gen II PWR fuel assemblies were expected to withstand several DPA[38]. In BWRs the materials are designed to be irradiated by  $10^{22} n/cm^2/s$ , experiencing approximately 7 DPA over the lifetime of the components[39]. Components in PWRs are irradiated by up to  $10^{22} n cm^{-2}s^{-1}$ , and are expected to operate up to an irradiation dose of 70 dpa[39]. To remain cost efficient, components in Gen IV reactors will be expected to operate safely up to even higher doses or irradiation damage (fig 3.5). Components in sodium-cooled fast reactors will be expected to withstand up to 200 DPA over their lifetime to meet the requirements of cost effectiveness and durability[38].

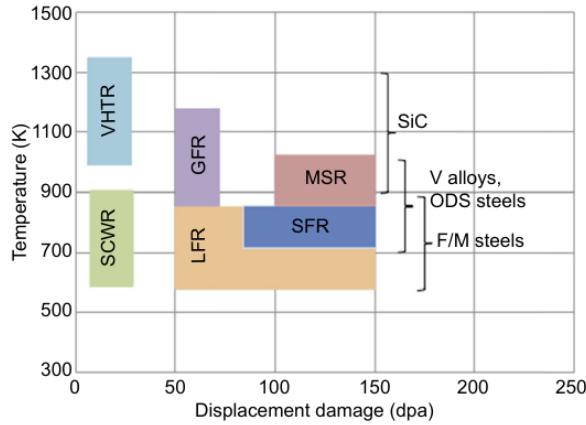


Figure 3.5: Expected DPA during component life time and operating temperatures[38]

Neutrons released during the fission of U235 range from 0 to 14 MeV (fig 1.9), and the energy transferred to a target atom depends on the scattering angle of the neutron and the mass of the target nucleus.

$$\alpha = \left( \frac{m_N - 1}{m_N + 1} \right)^2$$

$$E_N = E_n \left( 1 - \frac{(1 + \alpha) + (1 - \alpha) \cos\theta_C}{2} \right) \quad (3.1)$$

In a collision, the amount of energy transferred to a target atom will depend on both the scattering angle of neutron recoiling from the target and the mass of the target nucleus. If the neutron does not change direction, a small amount of energy will be transferred, but if it bounces back at 180 degrees it will transfer the maximum amount of energy. A neutron will lose more energy per collision with light atoms (hydrogen, helium, carbon), but much less energy per collision with larger atoms (iron, molybdenum, lead).

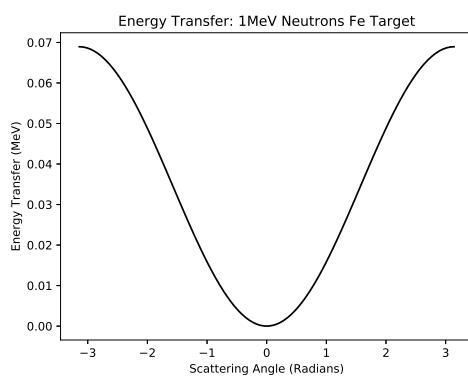


Figure 3.6: Scattering angle probability of neutrons

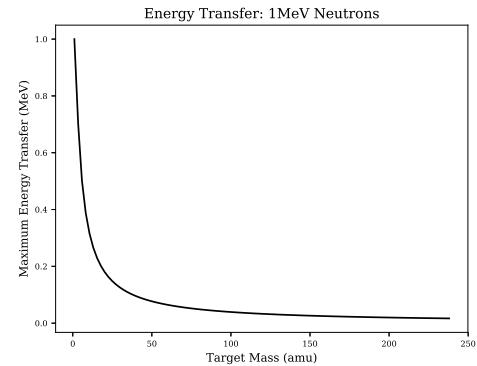


Figure 3.7: Energy transfer as a function of target mass

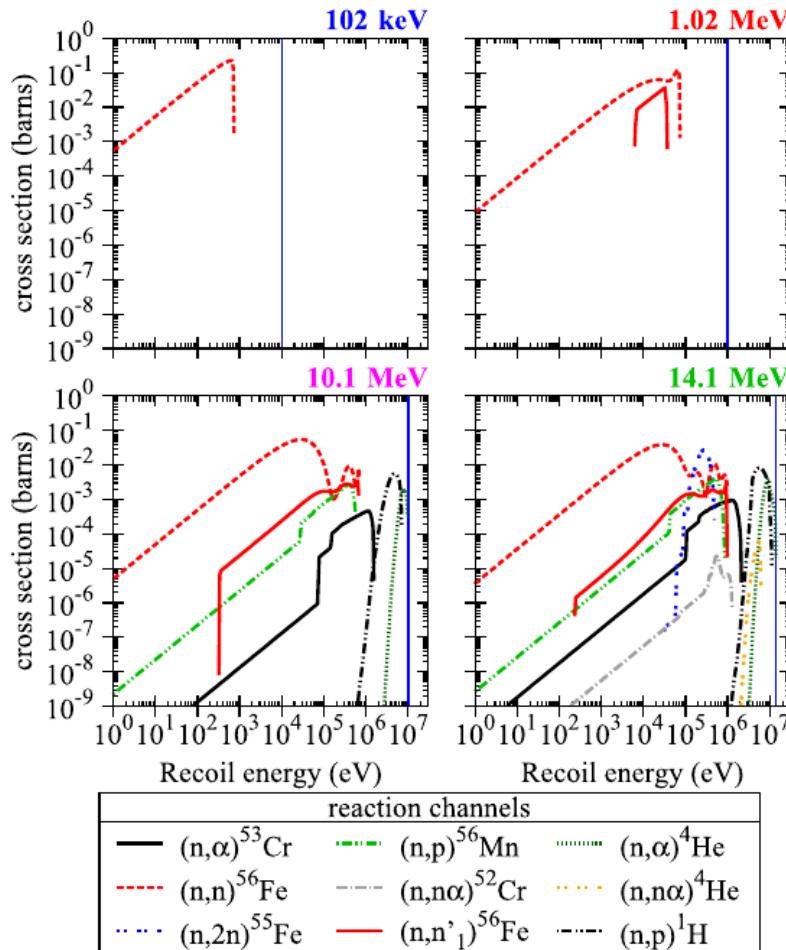


Figure 3.8: Recoil energies for neutrons at 102keV, 1.02MeV, 10.1MeV and 14.1MeV[23]

The energy of Fe PKAs as a result of neutron irradiation has been calculated for an Fe target[23] (fig. 1.10), and those created by 1MeV neutrons may have an energy up to 100keV. As nuclear reactions are also possible between the neutrons and the target nuclei, there is an inelastic  $(n, n)^{56}Fe$  channel as well as other elastic scattering channels where the post collision particles are different to the pre collision particles (fig. 3.8).

### 3.2.4 Swelling

Swelling results from damage to the crystalline structure of the metal, and is formed through small voids being created in the crystal. As vacancies are created by radiation damage, they precipitate into small voids. The temperature range for such swelling is primarily from 670K to 870K.

Materials have been known to swell to a 100% increase in their original volume at these intermediate temperatures under irradiation[40]. Swelling is sensitive to the total damage amount, the temperature and composition of the alloy. An increase in Ni decreases swelling as does the addition of Zr. Small amounts of P and Mo (0.02% and 0.5-1.0% respectively) cause the swelling of the alloy to increase, however higher concentrations of either cause swelling to decrease[41].

As a component swells its properties change. By definition the volume changes, but so does the elastic moduli. Not only this, but the swollen component may stress itself and surrounding components, and this may lead to SCC.

In the EBR-II structural 304SS was irradiated to 20DPA. At a temperature of approximately 650K the steel's volume increased by 2% due to swelling[42]. With components in future reactors expected to withstand ten times this damage, swelling is a concern. However, increased temperatures and a careful balance in the composition of future alloys will help to reduce this.

### 3.2.5 Radiation Induced Segregation

The diffusion of atoms within an alloy has an impact on the characteristics of the metal [43]. When radiation causes point defects, these interstitials also diffuse parallel to thermal solute diffusion. At low temperatures, the atoms are unable to diffuse at an appreciable rate; the mobility of vacancies are low[34] and there are an excess of vacancies due to radiation damage, and this leads to recombination of defects. At high temperatures, there is a higher concentration of thermal defects[39]. This increases the defect recombination rate and reduces migration of defects to sinks, such as grain boundaries.

The melting temperature range of 304 and 316 stainless steel are approximately 1695-1722K and 1644-1672K respectively. Gen II reactors, such as Sizewell B, have an operating temperature of several hundred degrees centigrade. The inlet temperature for the Sizewell B PWR reactor is 566K[44] and the outlet temperature is 597K[44]. Operating at approximately 35% the melting point of the steel this is, unfortunately, a good temperature for radiation induced segregation (RIS) to occur.

The Kirkendall effect (KE) concerns the diffusion of one material into another and vice versa at an interface. In the original experiment performed by Kirkendall and Smigelskas, brass (Cu and Zn) was sandwiched between Cu and left at a temperature of over 1050K for almost two months. Using Mo as a marker, it was discovered that Zn diffused out of the brass faster than the copper diffused into the brass. The flux of atoms results in a flux defects and a shrink in volume of brass[45].

The inverse Kirkendall effect (IKE) is driven by an external force, such as irradiation. Irradiation of the material causes a flux of defects and, inverse to the KE, this causes a flux of atoms. For an austenitic stainless steel the diffusion coefficients in order of size are  $D_{Cr} > D_{Fe} > D_{Ni}$ [46], so Cr will diffuse away from grain boundaries and other sinks, followed by Fe. As Ni is slowest to diffuse, this will be enriched. An additional mechanism that is thought to contribute to RIS is interstitial flux where undersized atoms as interstitials are preferential[46].

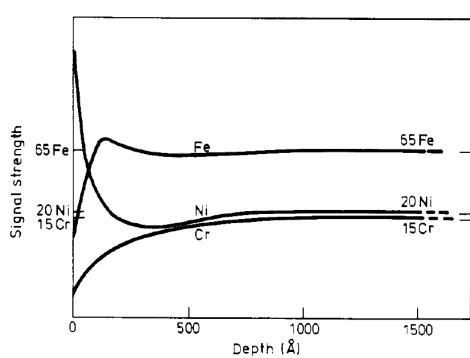


Figure 3.9: Depletion of cr, fe and enrichment of ni at the surface[47]

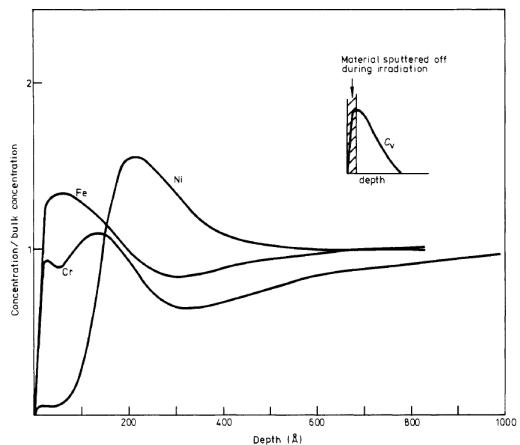


Figure 3.10: Concentration profile after 46DPA irradiation with 75 keV Ni ions[48]

Experimental work by Johnston et al irradiated steel, containing 15% Cr and 20% Ni, with 4MeV Ni ions. It was irradiated to a damage dose of 8DPA at 950K. Some material will have been sputtered from the surface, but in the remaining material Cr, and Fe, were depleted. Ni, however, was enriched (fig. 3.9).

In work by Marwick a similar alloy with 14% Cr and 15% Ni was irradiated to a much higher dose of 46 DPA with lower energy 75KeV Ni ions. In this instance the damage occurs in a much thinner layer of the surface that was approximately 30nm thick (fig. 3.10). Conversely to Johnston et al, in this thin layer there is a depletion of Ni and an enrichment of Cr in the first 20nm or so. However, between approximately 25nm and 60nm there is a similar depletion of Fe and Cr as well as an enrichment of Ni.

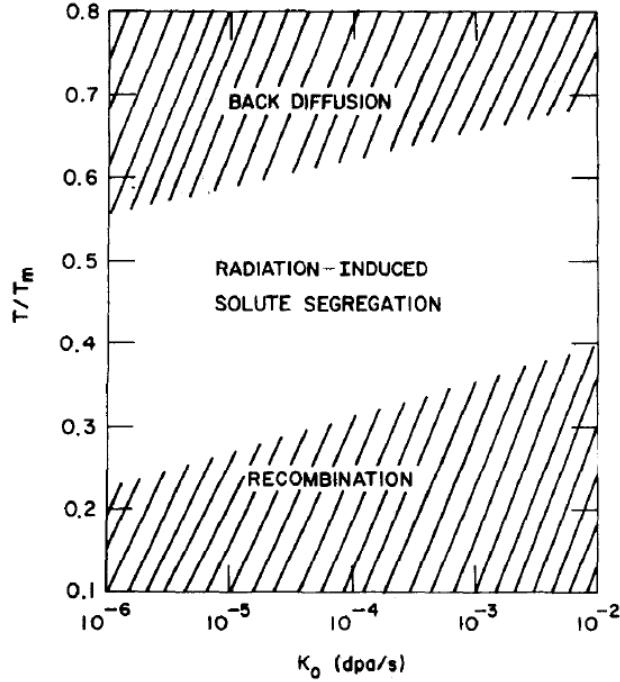


Figure 3.11: Temperature and DPA dependence of RIS[49]

The underlying mechanisms of RIS are dependent on both the temperature (relative to the melting point of the material) and the amount of damage (fig. 3.11). The result of RIS is a breakdown in the passive protection of Cr, which leads to IGSCC under the correct conditions.

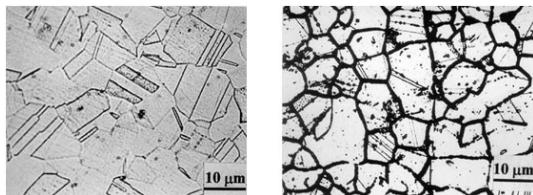


Figure 3.12: Formation of chromium precipitates at the grain boundary [54]

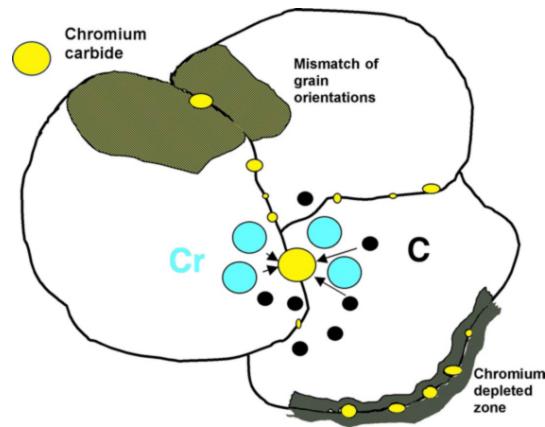


Figure 3.13: Sensitization of an alloy: chromium carbide precipitation [55]

### 3.3 Corrosion Resistance of Austenitic Stainless Steels

#### 3.3.1 Passive Film Protection of Chromium

The addition of Cr improves the resistance of stainless steel to corrosion by orders of magnitude. A very thin passive oxide layer, 20-30 angstroms in width, forms when the surface is exposed to an environment containing oxygen. The protective layer is self repairing and, if the surface is damaged, the oxide layer reforms in the presence of oxygen.

When the steel is in an environment containing oxygen, electrons within the metal tunnel through the surface. As an oxidizing agent, the nearby oxygen atoms readily accept the electrons. A strong electric field forms between positive ions within the metal and the negatively charged oxygen atoms [50].

Fe at the surface of the forming oxide layer is preferentially dissolved away over Cr [51] and Cr ions within the oxide layer have a lower mobility than Fe ions [52]. Ni remains in place within the alloy, and this leads to an enrichment of chromium oxide in the passive layer. Once the layer is thick enough to reduce the electric field across it, the formation of the layer stops. As mentioned previously, this is at a layer thickness of 2-3nm for stainless steel.

The  $\text{Cr}_2\text{O}_3$  layer may be formed by heating the steel to 770K[53]. However, at higher temperatures, the steel begins to lose its protective layer due to sensitization.

#### 3.3.2 Sensitization and Passive Film Removal

Steel by definition is Fe alloyed with varying small percentages of C. C changes the property of the alloy, and one example of this is an increase in hardness over pure Fe. At elevated temperatures, 670K to 1020K, the added Cr within the steel forms precipitates of Fe-Cr carbides  $(\text{FeCr})_{23}\text{C}_6$  at the grain boundaries, reducing the percentage of Cr at the grain boundary and removing the layer of passive protection.

The sensitization of the steel can be reversed. By heating the steel further, to approximately 1,320K to 1,420K, the steel is solution annealed dissolving the carbides back into the steel.

#### 3.3.3 Addition of Molybdenum

A common austenitic stainless steel is 304SS and this relies heavily on the passive film due to a high content of Cr, ranging from 17.5% to 20% for grades 304, 304L and 304H. A similar stainless steel, 316, has slightly less Cr, slightly more Ni and 2-3% Mo, an element not present in 304 stainless steel.

Pure Mo has a BCC crystal structure, and it is a ferrite former. However, high proportions of austenite former such as Ni and N force 316 stainless steel to keep its FCC structure despite the addition of Mo. Adding it to austenitic steels improves their strength and resistance to creep at high temperatures.

Steels such as 316SS are known to have better corrosion resistance in chloride rich environments. Where at least 2% Mo has been added it improves resistance to pitting and crevice corrosion resistance[56]. The mechanism by which Mo helps to protect against corrosion is still unclear. It is possible the addition helps to reduce the breakdown of the passive film protecting the steel, but it may also be the case that Mo promotes the repair of the passive film [57].

### 3.4 Irradiation Assisted Stress Corrosion Cracking

Damage due to irradiation was first identified in high stress stainless steel components in a reactor, such as bolts, springs, and fuel elements[58][59], and later being found in lower stress austenitic stainless steel components[59].

Irradiation of water causes the radiolysis of water, changing the chemistry of water and creating hydrogen, radicals and other ions that increase the corrosiveness of the environment (fig 3.14). The radiation also damages the material directly, changing its properties. A form of IASCC to which high nickel alloys, including austenitic steels, are particularly susceptible to is inter granular stress corrosion cracking.

### 3.5 Inter Granular Stress Corrosion Cracking

Stress corrosion cracking may be trans granular (through the grains) or inter granular, at the grain boundary. IGSCC is a particularly prominent failure mechanism for austenitic stainless steels. For IGSCC to occur, there must be a material that is susceptible to this form of cracking, an environment that is corrosive to the material as well as stress (fig. 3.15).

Stress may come from a high pressure environment or residual stresses due to welding. In a nuclear reactor the atomic structure may be under stress due to damage caused by neutrons passing through the steel, or due to the swelling of the steel on the macroscopic scale. If Cr is depleted at the grain boundary, protection to corrosion due to the passive layer is lost. This, coupled with the knowledge that austenitic stainless steels are susceptible to IGSCC, completes the three requirements and, over time, components of this material in these conditions will eventually fail to IGSCC.

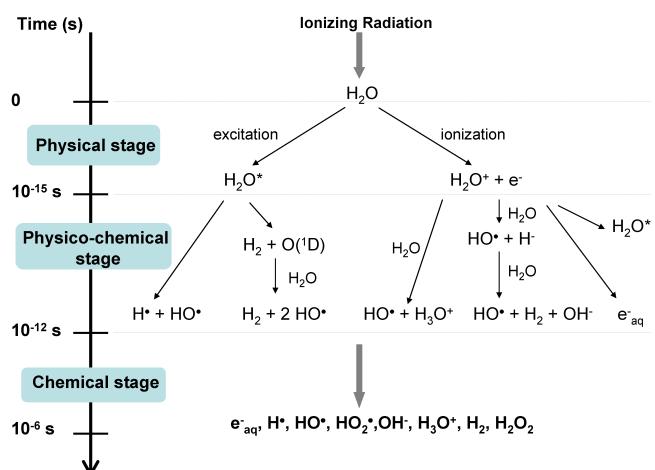


Figure 3.14: Changing water chemistry: radiolysis of water[60]

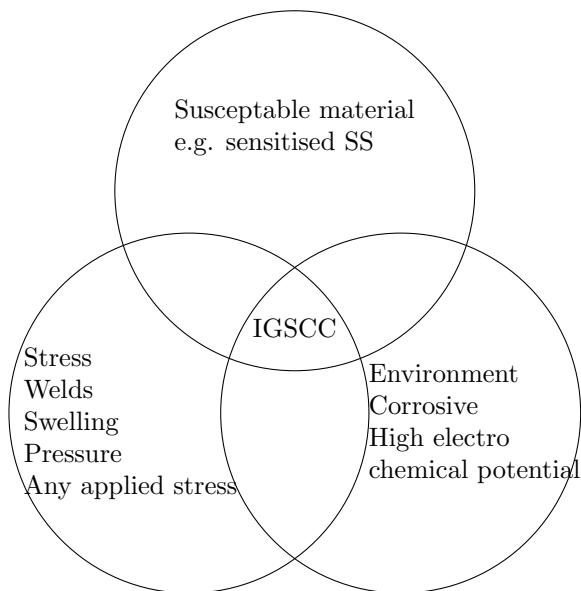


Figure 3.15: Three requirements for IGSCC to occur

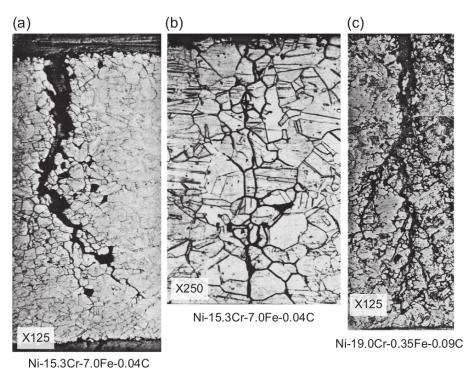


Figure 3.16: Inter Granular Stress Corrosion Cracking in Nickel Alloy[61]

IGSCC has not only been a defect in stainless steel. High Ni content alloys, such as Alloy 600 (Inconel: Ni-72, Cr-17, Fe-10), have been known to suffer from IGSCC since the very early days of nuclear energy, in particular with the prototype S1W reactor, prototype for the first nuclear powered submarine, the USS Nautilus.

After the war, there was a drive to develop a nuclear powered navy for the US. There are obvious benefits in replacing conventional power in vessels with nuclear, in particular for submarines. This was pushed by Admiral Rickover and the route to building the USS Nautilus began. A number of available Fe-Cr-Ni alloys were considered for the construction of the reactor (fig. 3.17).

Prototypes developed for the USS Nautilus experienced stress corrosion cracking in the Inconel Alloy 600 components. H. Coriou replicated this damage to Inconel in the laboratory, holding a sample at 620K in deoxygenated water for 3 months[61]. Further studies showed that IGSCC became a particular issue to alloys with a high Ni content in pure water, commencing with a content of approximately 70% Ni. In a weak salt water solution (0.1% NaCl) IGSCC occurred as with the same alloy in pure water, but TGS SCC occurred in similar alloys with a low Ni content (less than 15%) (fig. 3.18).

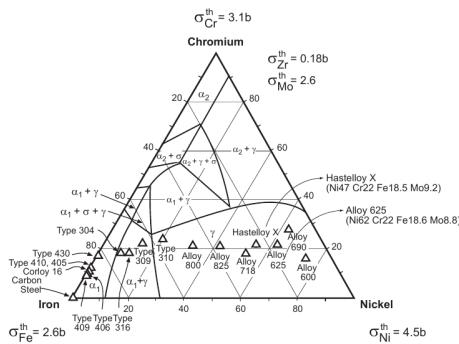


Figure 3.17: Alloy choices for early LWRs

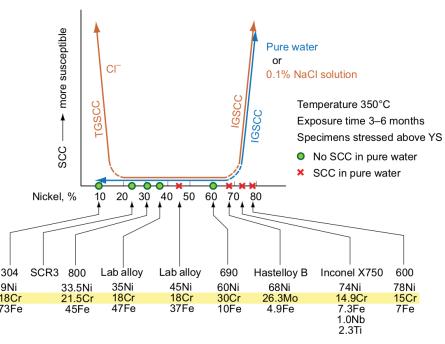


Figure 3.18: Nickel, IGSCC and TGSCC

### 3.5.1 IGSCC in Light Water Reactors

In LWRs a major factor in corrosion of components is the water chemistry and electrochemical corrosion potential that they operate in. Higher levels of oxygen dissolved in the water leads to an increase in the corrosion potential[40]. Other factors within LWRs include depletion of Cr due to either heat treatment, welding or irradiation.

The primary circuit of a BWR includes components such as a reactor pressure vessel, piping that leads out of the containment building, fuel and fuel assembly. 300 series steels, including 304 and 316, are used within the primary circuit, including the piping and control rod absorbers. In BWRs the purity of the water has been addressed, and the addition of hydrogen to the water has reduced cracking of components[61].

Higher nickel content steels such as the 600 series alloys are used in PWRs, and components that are made from these alloys include steam generators and pressure vessels. In the AP-1000 design, the control rod drive mechanism at the reactor coolant pressure boundary are made from 304, 304L, 304LN, 316, 316L and 316LN steel. Higher nickel content 690 is used for penetration into the pressure vessel[62]. As illustrated by Coriou's work in the 1950s and 1960s, higher nickel content steels are susceptible to stress corrosion cracking.

### 3.5.2 IGSCC and Advanced Gas-cooled Reactors

The first generation of nuclear reactors in the UK were Magnox type reactors. They used un enriched Uranium 0.7% U235 as fuel which was contained within magnesium oxide cladding due to its low absorption cross section. A major drawback was the relatively low operating temperature of 630K.

Carnot's theorem shows that the maximum amount of energy from an engine is dependent on the difference between the hot and cold reservoirs (section 1.3.4). The ambient temperature of the power station will vary a small amount with the seasons. The practical way to improve the maximum possible efficiency is to increase the engine (reactor) temperature.

$$\eta_{max} = 1 - \frac{T_c}{T_h} \text{ where the temperature is measured in absolute units} \quad (3.2)$$

AGR reactors were designed to run at much higher temperatures of  $920K$ . By increasing the temperature, the maximum possible thermal efficiency was increased from just over 50 percent to almost 70 percent. To withstand higher temperatures, the magnesium oxide cladding used in the earlier Magnox reactors was replaced with stainless steel. To counteract the higher neutron absorption cross section of the cladding, the fuel was enriched up to 3.5%  $U^{235}$ .

The cladding holds the fuel elements together and at the required location inside the reactor whilst it is consumed in the reactor. The cladding also performs several functions once the fuel has been spent. It holds the fuel elements together and acts as a primary containment for the spent fuel[63].

During its lifetime as a fuel element within the reactor, the cladding has been heated and irradiated. By the process of RIS the Cr at the grain boundary can drop to 10% concentration[64]. The C rich environment within the reactor, due to the CO<sub>2</sub> coolant, provides yet another mechanism to deplete Cr at the grain boundary by the formation of Fe-Cr carbides (FeCr)<sub>23</sub>C<sub>6</sub>.

This loss of protection at the grain boundary, stress due to the role of the component within the reactor, the changing environment due to radiation damage and the material's susceptibility lead to IGSCC.

## 3.6 Austenitic Stainless Steels in Gen III+ and Gen IV Reactors

The AP1000 is an advanced PWR. As a Westinghouse reactor, the fuel cladding will be their trademarked Zirlo alloy. Inconel will be used for the steam generator and heat exchanger with carbon steel used for the construction of the pressure vessel. The control rod absorbers, however, will be constructed from 304SS[65].

Areva designed the EPR and this will use 316SS as the fuel cladding material. It will also use the slightly less corrosion resistant 304SS (forged) in parts of the control rod drive mechanisms[66].

The perforated upper core plate, between the reactor core and the inlet/outlet/control rod mechanisms, is also made from austenitic stainless steel. The pressuriser is constructed from ferritic steel, but its internals are clad with austenitic stainless steel to protect the structure from the coolant.

Gen IV designs also include the use of austenitic stainless steels. Experimental fast reactors in many countries, including the US, UK, Japan, France, India and China have used either 304SS, 316SS or both in their construction.

Austenitic steels are more corrosion resistant than ferritic or martensitic steels. They do expand more as a result of being irradiated, but they are stronger at higher temperatures. Many of the Gen IV reactors operate at much higher temperatures than existing reactors and the components will be expected to resist higher doses of radiation damage throughout their lifetime.

GFRs are expected to use austenitic steels. Due to their chemical compatibility with sodium they are also expected to be used in SFRs. As well as being used in the reactor core, these steels will also be used outside the core. For example, SFRs in France will use austenitic steels in the secondary circuit, primary pump, internal heat exchanger and more[67].

## 3.7 The Addition of PGMs to Stainless Steel

Water purity is a concern for light water reactors. The fewer contaminants there are to begin with, the better. Even so, the radiolysis of the purest water will still create a steady supply of corrosive molecules (fig. 3.14). Adding hydrogen to the water reduces the electrochemical corrosion potential of the water, but if too much is added it will combine with the radioactive nitrogen-16 created by the  $^{16}O(n, p)^{16}N$  reaction to form  $^{16}NH_3$ [68].

Cathodic modification is a technique used to improve corrosion resistance, and one way to do this for stainless steel is to add small amounts of platinum group metal (PGM)s[69]. By adding such metals, the anodic reaction may be reduced. PGMs also act as a catalyst in the reduction and removal of O<sub>2</sub> and H<sub>2</sub>O<sub>2</sub> from the water[68].

The increased corrosion resistance is dependant upon the amount of PGM added, but a higher percentage does not necessarily mean better protection. This is important for two reasons, to find the optimum amount for the

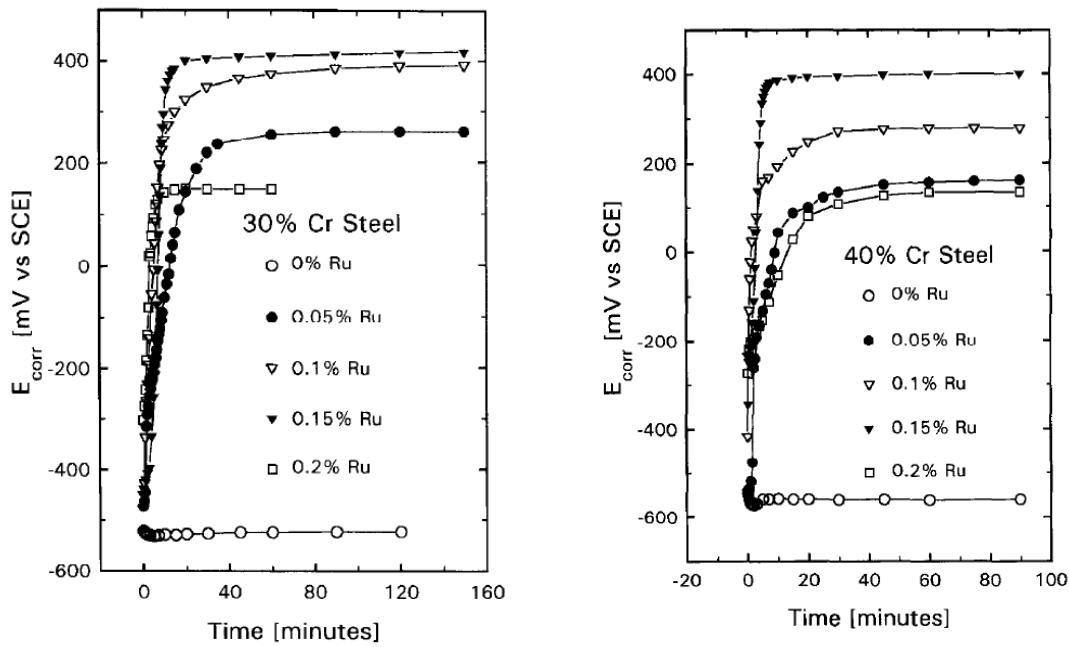


Figure 3.19: Addition of Ru to Fe30Cr and Fe40Cr steels - open current corrosion potential variation with time in 10% sulphuric acid[69]

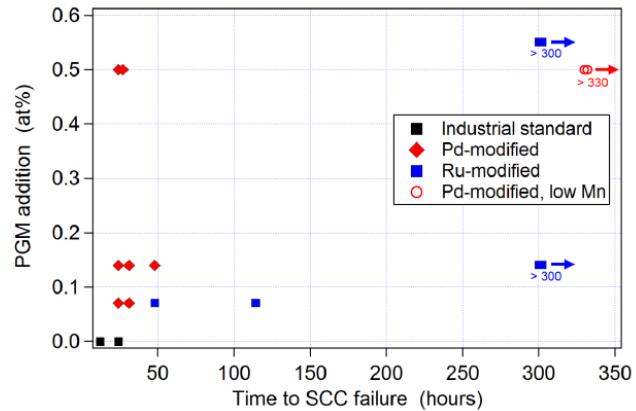


Figure 3.20: Failure times for 304SS, Pd doped 304SS (with and without Mn) and Ru doped 304SS in polythionic acid at ambient temperature[1]

sake of corrosion resistance, and to keep the price down as the additions are so expensive, even in such small amounts.

It was found that an addition of 0.15% Ru (out of a selection of 0%, 0.05%, 0.10%, 0.15% and 0.20%) was optimum for both Fe30Cr and Fe40Cr steel (fig. 3.19)[69]. Other PGMs may be used for cathodic modification, including Pd, Ir and Pt, but the cost of the metal must be weighed against the possible gains.

Connolly et al recently added Ru and Pd to 304SS samples that were sensitized at 920K for 24 hours[1]. After sensitization, the steel was analysed by TEM and Cr-rich carbides were found at grain boundaries. Without the passive protection due to Cr, the samples were exposed to polythionic acid to promote SCC. The Ru doped steels performed well, but the Pd doped steels showed no improvement over regular 304SS. Further analysis of the samples by TEM reveals that the low percentage of Mn in the steel allows the formation of Pd-Mn precipitates at the grain boundary[1]. These precipitates form during the sensitization procedure and thus negate the protection by cathodic modification at the surface of the metal by removing the Pd.

With a low Mn 304SS (less than 0.05% Mn), varied amounts of Pd have been tested. With the reduction in Mn

the corrosion resistance improves greatly over regular 304SS. Palladium remains at the surface, as the amount of Mn is too low for appreciable amounts of Pd to be lost to precipitate formation during sensitization. No failures have been observed with either the Ru or Pd (low Mn) samples for 330 hours exposure to polythionic acid at ambient temperatures (fig. 3.20) [1].

The surface of the metal does not need to have a full coating of PGM but it should be equally spaced across the surface. Oxidant in the layer at the surface will be consumed at the locations of PGMs and the change in concentration at those points will naturally cause diffusion of more oxidant to those locations[68].

The PGMs may be coated on the surface of the metal or alloyed to the metal. The first method is the more cost efficient but if a crack does form, or if a layer is removed, it will expose the regular alloy underneath. If the PGM is alloyed, there will always be protection, unless a mechanism is removing the PGM from the surface. This could be through precipitation as a carbide, or it could be due to radiation causing the metals to segregate.

Reaction	Product Halflife
$^{54}Cr(n, \gamma)^{55}Cr$	3.5 mins
$^{64}Ni(n, \gamma)^{65}Ni$	2.5 hrs
$^{55}Mn(n, \gamma)^{55}Mn$	2.6 hrs
$^{104}Ru(n, \gamma)^{105}Ru$	4.4 hrs
$^{108}Pd(n, \gamma)^{109}Pd$	13.7 hrs
$^{196}Pt(n, \gamma)^{197}Pt$	19.9 hrs

Table 3.2: Radioactive products and their half lives for PGM doped 304SS

Adding elements to metals, even in small amounts, may be problematic as certain elements have high neutron reaction cross sections and transmute into radioactive isotopes.  $^{55}Cr$  is a concern for all neutron irradiated stainless steels, but with a very short half life it quickly cools and becomes less of a concern, with  $^{65}Ni$  becoming the dominant source of activity after 10 hours. Adding Mn to steel that is in a reactor is known to result in an increase in activity due to the  $^{55}Mn(n, \gamma)^{56}Mn$  reaction (section 2.4).

The addition of PGMs alters the activity of the steel following neutron irradiation and with just a 1% addition there is an increase in activity comparable to that of the 10% Ni transmuted to  $^{65}Ni$ . This is due to reactions that create  $^{105}Ru$ ,  $^{109}Pd$  and  $^{197}Pt$  for Ru, Pd and Pt respectively (table 3.2).

With such small additions required for cathodic modification and the short half lives of the radioactive isotopes, there isn't a concern due to neutron activation over regular 304SS, and much less of a concern than that of 304SS containing Mn. More data computed using the TENDL library is available in appendix ??.

# Chapter 4

## Background: Proton Activation and Radioactive Decay

*Where proton or deuteron radiation is used to emulate neutron damage, there is still an associated risk of transmuting atoms in the target material, creating radioactive isotopes. An equation was derived to calculate the radioactivity of each element in the decay chain. It takes into account a source term for each isotope in the decay chain, as well as any branching factors.*

*Once derived, the equation was used in a computer code that, coupled with cross section data from the TENDL library and ion transport data from the SRIM ion transport code, predicts the radioactivity of a sample (chapter 7).*

### 4.1 Ion Sources

#### 4.1.1 LINAC

Since the development of the first linear accelerator (LINAC) in the 1940s, their modern day versions have become some of the most powerful accelerators in the world. The longest LINAC, Stanford Linear Accelerator Center, 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.

#### 4.1.2 Cyclotron

Cyclotrons are reasonably compact and cost effective. The largest current cyclotron, TRI-University Meson Facility, is located in Canada and is able to output protons with energies over 500MeV. It is relatively large, weighing 170 tons. There are approximately 350 cyclotrons[70] around the world today.

The University of Birmingham cyclotron is more compact. The protons it accelerates are in the 8-40MeV range and the projectiles are not just restricted to protons (table 4.1).

Projectile	Energy (MeV)	Maximum Current (micro A)
proton	8-40	60
deuteron	8-40	30
$He^{2+}$	8-53	30

Table 4.1: University of Birmingham Cyclotron Ion Beams

The moderate size, energy range, availability and cost of these Cyclotrons make them ideal candidates for damaging targets with light ions, rather than neutrons from a neutron source.

The Scanditronix MC40 is an isochronous cyclotron. As the ions are accelerated to higher and higher velocities, the effects of special relativity become appreciable.

$$v = c \sqrt{1 - \frac{E_{rest}^2}{E_V^2}} \quad (4.1)$$

The cyclotron has two Dees and these are D shaped hollow electrodes. Ions enter at the center of the cyclotron and are held in the Dees as they are accelerated. The magnetic field is varied to bend the path of the ions from a circle into a round cornered triangle.

### 4.1.3 Synchrotron

Two of the most well known accelerators are Synchrotrons: the Large Hadron Collider at Conseil Européen pour la Recherche Nucléaire, and the now retired Tevatron at Fermilab. These are typically large machines and there are approximately 70 around the world[71], approximately a fifth the number of Cyclotrons.

Synchrotrons are used for storing high energy particles in a continuous loop, either to generate light through magnetobremssstrahlung, or for high energy collisions. Cyclotrons are more commonly used for lower energy (tens of MeV) projectiles, in materials science for damaging materials and for medical use (creating radioactive isotopes for use in hospitals).

## 4.2 Neutron Sources

### 4.2.1 High-Flux Neutron Reactors

There are 250 or so research reactors in 55 countries [72], 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[73]. 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 [72].

#### 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

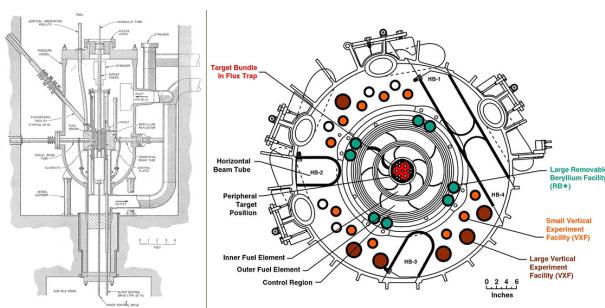


Figure 4.1: A cross section of the HFIR [75]

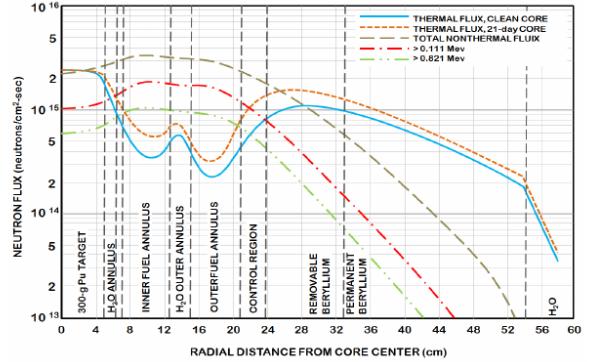


Figure 4.2: Neutron Flux in the HFIR while at 85MW [76]

highly enriched Uranium as its fuel source and is scheduled to operate at 100% capacity for 161 days per year, in cycles of approximately 23 days. A 30cm surround of Beryllium is used to reflect neutrons, and in the reactor core there is a high flux of thermal neutrons at a rate of  $2.3 \times 10^{15}$  neutrons  $\text{cm}^{-2}\text{s}^{-1}$ [74].

### NIST Center for Neutron Research

The National Bureau of Standards Reactor (fig. 4.3) was designed as a research reactor with a power output of 40MW and a neutron flux of approximately  $1.0 \times 10^{15}$  neutrons  $\text{cm}^{-2}\text{s}^{-1}$ . It uses highly enriched Uranium as fuel and is both moderated and cooled by heavy water.

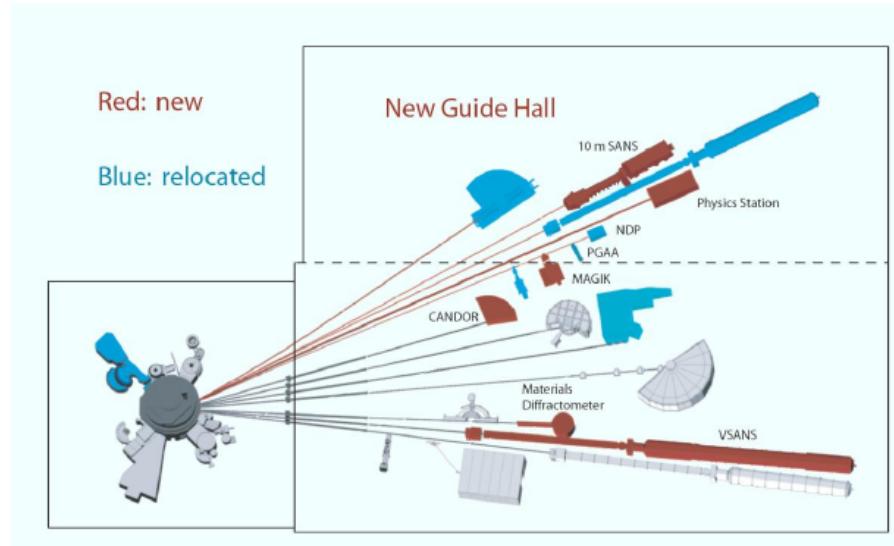


Figure 4.3: NBSR building layout[77]

#### 4.2.2 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 is a neutron spallation source located in Harwell, Oxford. Protons are accelerated in a LINAC to 0.37c before being fed into a 163m circumference synchrotron. The protons are then accelerated to 0.84c before being projected, in bunches, at a tungsten target. The impact with the tungsten releases neutrons[78].

Oak Ridge National Laboratory also has a neutron spallation source, the SNS. A LINAC accelerates a negatively charged proton (a hydrogen atom with two electrons) up to just below 0.9c. It is stripped of the electrons and enters an accumulation ring as a proton. More protons are accumulated until the beam in the accumulator is diverted at a mercury target. The collision between the mercury target and the protons releases approximately 20 neutrons per collision, and these neutrons pass through beam lines to the experiments.

The source creates a pulse of neutrons from the 50 ton mercury target 60 times per second. The protons are accelerated by the LINAC to between 2.5MeV and 1.0GeV and this results in neutrons with energies almost as high as the proton projectiles. A spectrum of neutron energies is produced with each pulse, although the neutrons may be moderated at the end of the beam lines to slow the neutrons down.

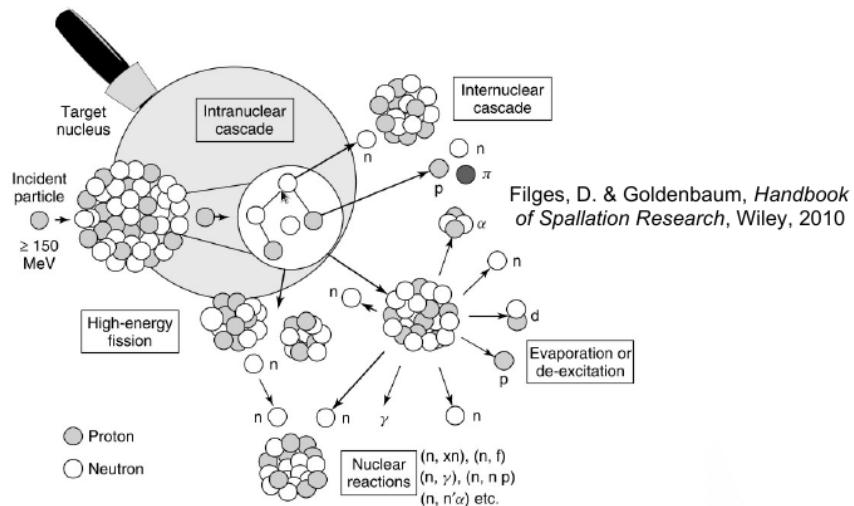


Figure 4.4: An example of a neutron spallation source where high energy ions collide with heavy atoms[79]

## 4.3 Source Review

Source	Cost	Projectile	Flux/Current	Energy
Scanditronix MC-40	£1 million	Proton	$3.7 \times 10^{14}$	8-40MeV
HFIR[76]		Neutron	$3.0 \times 10^{15}$	Full Spectrum
HFIR[76]		Thermal Neutron	Over $2.0 \times 10^{15}$	Thermal
HFIR[76]		Fast Neutron	$2.0 \times 10^{15}$	$> 0.111\text{MeV}$
HFIR[76]		Fast Neutron	$1.0 \times 10^{15}$	$> 0.821\text{MeV}$
SNS[80]	\$1.4 billion	Neutron	Average $1.2 \times 10^{13}$	Full Spectrum
ISIS spallation source[80]		Neutron	Average $4.0 \times 10^{13}$	Full Spectrum

Table 4.2: Examples of neutron sources, energies and flux, with the MC-40 as a reference for protons

The compact proton source that is the cyclotron is compared to neutron sources. It is much smaller and cheaper in comparison and the energy of the projectile may be controlled, whereas a spectrum of energies of neutrons is produced by reactors and spallation sources. Faster neutrons will cause more damage to materials in Gen IV reactors giving the cyclotron an advantage, being able to focus in just one energy range. The three mentioned isotope sources are national facilities that cost much more than a cyclotron. Access is shared between many researchers whereas a cyclotron, such as the Scanditronix MC-40 at the University of Birmingham may have a beam dedicated to a particular task.

## 4.4 Ion Irradiation to Investigate Neutron Damage

Neutrons emitted during the fission of Uranium-235 have an energy spectra in the intermediate to fast range, with a peak at 1MeV, and a large proportion in the 1MeV to 10MeV range (fig. 4.5). The higher energy neutrons are more of a concern to this work as higher energy neutrons, on colliding with atoms within the target material, cause large damage cascades.

As discussed earlier in this chapter, there are a number of methods available to create both high energy neutrons and ions, but each has its own set of advantages and disadvantages. At the University of Birmingham high energy ions are created using a cyclotron.

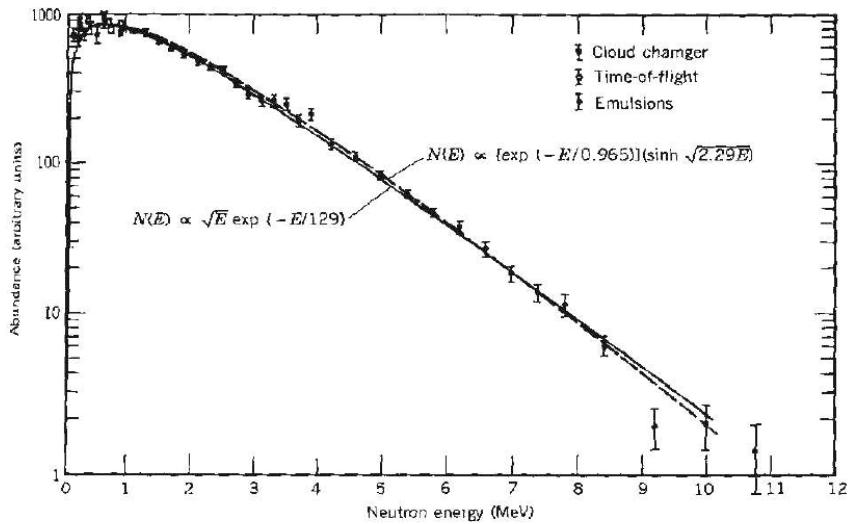


Figure 4.5: Neutron Spectra from the Fission of U235[81]

#### 4.4.1 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 ( $3.1 \times 10^{14}$  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 reaction cross section for neutrons also cover a much larger range including lower energy projectiles, something Coulomb repulsion reduces for ions (fig. 4.6).

The Activity code discussed in this work was developed to calculate the activity of a target material irradiated

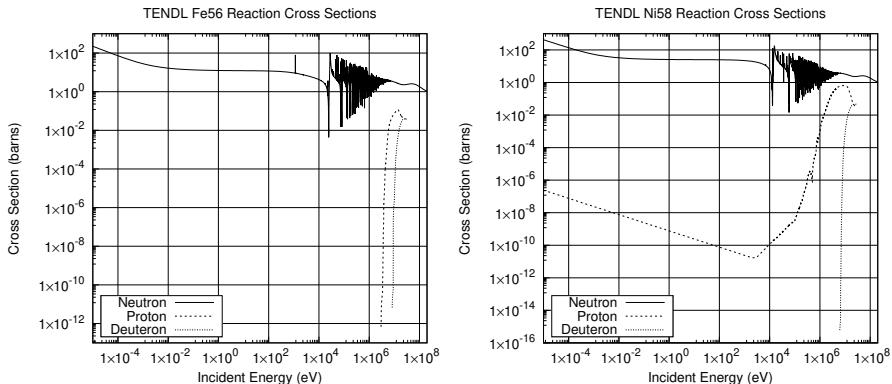


Figure 4.6: Neutron, proton and deuteron cross sections for Fe56 and Ni58

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 Nuclear Data Services radioactive decay database. A second version was also developed using Python and a more recent TENDL database.

#### 4.4.2 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 moderators and coolants, such as water. At lower energies the neutrons may be captured by the nuclei of atoms they interact with. This creates a new isotope which may or may not be stable.

##### 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. The barrier between a proton and a target nucleus may be approximated using eq. 4.2[82] and eq. 4.3[83]. Using these equations does neglect the probability of lower energy protons being captured via quantum tunnelling of the Coulomb barrier.

$$V_{Coulomb} = \frac{e^2}{4\pi\epsilon_0} \frac{Z_1 Z_2}{r_1 + r_2} \quad (4.2)$$

$$r = r_0 A^{1/3} \text{ where } r_0 = 1.2 \times 10^{-15} m \quad (4.3)$$

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.

$$R = \frac{J}{Q} \cdot n_\rho \cdot \sigma \cdot 10^{-28} \delta t \quad (4.4)$$

The reaction rate is calculated from the microscopic cross section, using eq. 4.4, and the parameters are as follows:

- R Reaction Rate (reactions per second)
- J Beam current (A)
- $n_\rho$  Number density of target (atoms per cubic metre)
- $\sigma$  microscopic reaction cross section (barns)
- Q projectile charge e.g  $1.602177 \times 10^{-19} C$  for a proton
- $\delta t$  target thickness (m)

### 4.4.3 Nuclear Reaction Cross Sections

#### Reaction Cross Sections

The type of reaction for an individual reaction cannot be determined, but the probability of that reaction happening may be measured and future events predicted. This data may be gathered experimentally, or it may be calculated using various models.

#### EXchange FORmat Data File

EXchange FORmat (EXFOR) is a standard format for storing experimental results from nuclear reactions[84]. It is also associated with an online database hosted by the International Atomic Energy Agency (IAEA) where nuclear reaction data from a variety of sources are obtained and compared. Thousands of points make up the database and the work of three groups responsible for subsets of that data was reviewed to give an insight into the methods used.

The (p,n) reactions with pure Titanium, Vanadium, Chromium, Iron and Nickel were studied by Tanaka and Furukawa[85]. The 63 inch cyclotron that was used by the Institute for Nuclear Study (INS) in Tokyo provided the protons. The device output a mono-energetic 14MeV beam of protons at a stack of target foils. The Ti, Fe and Ni foils were  $2 - 5\text{mg/cm}^2$  in thickness, which is roughly 2-10 micrometers (depending on the density of the material). As the ions passed through the stack of foils their energy dropped and the experimenters estimated the energy using the Bethe stopping power formula[86]. The ion energies in the resulting data range from 3-4MeV up to 14MeV.

Similarly the (p,n) reaction in a range of materials with 5 to 10.5MeV protons was investigated by Wing and Huizenga[87]. A 60 inch cyclotron was used in this instance with foils of Vanadium, Chromium, Copper, Silver, Cadmium and Lanthanum. Foils were again stacked and had a thickness of  $5\text{mg/cm}^2$ , or approximately 5 micrometers, depending on the density of the element. Some energies as they passed through the foils were estimated by the experimenters, based on past work.

Vanadium, Iron and Copper foils were irradiated with 10 to 45MeV protons using the Milan Azimuthally Varying Field (AVF) cyclotron. In this work Gadioli et al[88] measured reaction cross sections for 17 proton energies. Thin foils ranging from 10-48 micrometers but is it unknown whether or not they were stacked or irradiated individually using aluminium absorbers.

The experimental work as a whole ranges over decades using slightly different tools and measuring methods, but the resulting data has contributed to ever improving models used to compute the probability of a reaction.

#### TALYS and the Optical Model Potential

In the standard model, protons and neutrons are composed of quarks, held together by the strong nuclear force. The nucleus of an atom is also held together by the strong nuclear force that on such small separations overwhelms the electromagnetic force of the protons with one another.

This is a complicated system, not to mention the excited states that nuclei may occupy. The interaction of a projectile (proton, neutron or a heavier ion) with the nucleus is a challenge to model.

Initially, reaction data for protons and neutrons were gathered through experimentation. In the 1950s, Feshbach et al put forward a simple model for nuclear reactions between the nucleus and neutrons, and this model was restricted to 0-20MeV neutrons. The form of the potential used was a complex function[89].

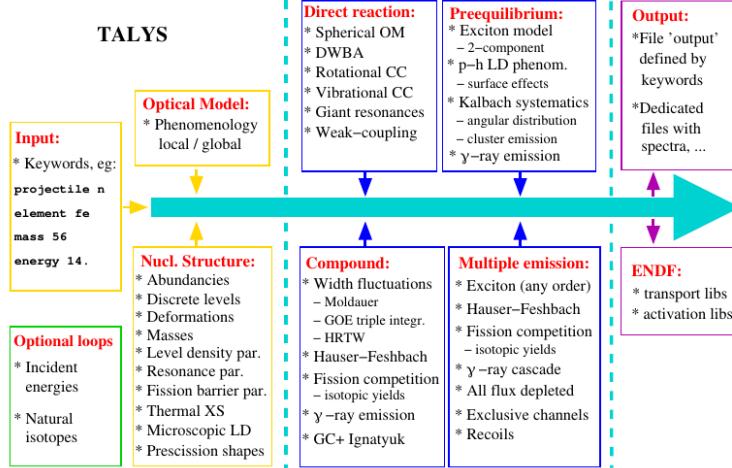


Figure 4.7: TALYS work flow [90]

$$\begin{aligned} V &= V_0(1 + i\zeta)r < R \\ V &= 0r > R \end{aligned} \tag{4.5}$$

The potential in equation 4.5 has the parameters  $R = 1.4 \times A^{\frac{1}{3}}$ ,  $V_0 = 42\text{MeV}$  and  $\zeta = 0.03$  where  $A$  is the atomic mass of the target atom.

Considering the complexity of the system being modelled, this simple model was very successful. Over the years since, the models used have become more complex and parameters used have been fit to an increasing amount of experimental data.

The Talys code uses a range of models (figure 4.7). These include the optical model which is solved using the Equations Couplées Itérations Séquentielle (ECIS-06) code of Jacques Raynal, implemented as a subroutine within Talys, and this is accurate up to 180MeV[90]. Whilst the Talys code has been extended up to 1GeV, it is experimental in the 180MeV to 1GeV range. Fortunately, this work only requires nuclear reaction cross section data up to approximately 100MeV as the current ion source under consideration produced ions with energies up to 60MeV.

The Talys code has potentials for protons and neutrons, but it also has potentials for deuterons, tritons, helium-3 and alpha particles. The potentials are discussed in detail in the Talys manual[90]. This extension to heavier ions may be useful for this work as the University of Birmingham cyclotron is capable of accelerating deuterons and  $He^{2+}$  ions.

Comparing experimental data to the Talys data for  $Fe54(p, \gamma)Co55$  shows good agreement between 1MeV and 10MeV4.8. There was insufficient experimental data from this source, but the Talys generated data ranges from micro eV up to 100MeV+ (fig. 4.8).

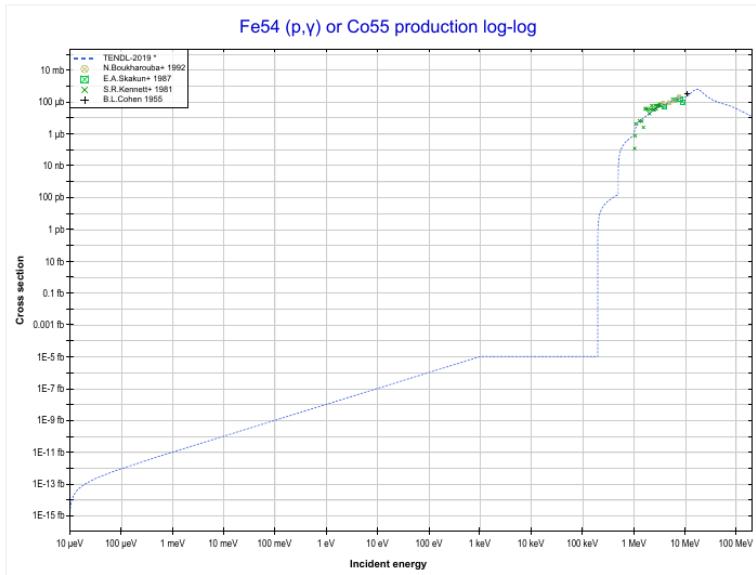


Figure 4.8: TALYS Fe54-Co55 cross section comparison with experiment [91]

## 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. The file contains the sum of all individual reactions as well as certain yields, and the data was generated using the TALYS and ALICE/ASH codes, as well as experimental data from Exfor[92].

Listing 4.1: A sample of the Iron-56 PADF data file

```

1   Proton Activation Data File           528 1451 12
2   ****
3   *****                                     528 1451 13
4   *****                                     528 1451 14
5   Authors: C.H.M.Broeders, U.Fischer, A.Yu.Konobeyev, L.Mercatali 528 1451 15
6   *****                                     528 1451 16
7   Data for PADF file were obtained using the TALYS code [1] for 528 1451 18
8   target nuclei with half-life more than 10 min and using the 528 1451 19
9   ALICE/ASH code [2] for nuclei with 1 sec < T1/2 < 10 min and 528 1451 20
10  available experimental data           528 1451 21
11  MAT numbers are taken according to JEFF-3.1/RDD      528 1451 22
12                               528 1451 23
13  Evaluation for Fe-56 (stable) : TALYS code          528 1451 24
14  Experimental data used for the correction of calculated 528 1451 25
15  excitation functions are taken from Refs.[3-12]       528 1451 26
16                               528 1451 27
17                               528 1451 28
18  File contains                         528 1451 29
19                               528 1451 30
20  MF=3 MT=5                           528 1451 31
21  Sum of all individual reaction cross-sections (Total 528 1451 32
22  reaction cross-section)            528 1451 33
23                               528 1451 34
24  MF=6 MT=5                           528 1451 35
25  Yields of nuclei in nuclear reactions including 528 1451 36
26  n, p, d, t, He-3, He-4 and photon production 528 1451 37

```

## TENDL 2019 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 TENDL nuclear reaction simulation code provides the calculated data.

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

Listing 4.2: Sample TENDL File

---

```

1 2.605600+4 5.545443+1      0      0      0      02631 3 2   1
2 0.000000+0 0.000000+0      0      0      1      462631 3 2   2
3      46      2                  2631 3 2   3
4 1.000000+3-9.920042-7 1.000000+6-9.920042-7 2.000000+6-2.167241-42631 3 2 4
5 3.000000+6-7.609736-3 4.000000+6-3.232345-2 5.000000+6-5.975554-22631 3 2 5
6 6.000000+6-5.149510-2 7.000000+6-3.685484-2 8.000000+6-5.542173-22631 3 2 6
7 9.000000+6-9.978765-2 1.000000+7-1.599116-1 1.100000+7-2.206871-12631 3 2 7
8 1.200000+7-2.743084-1 1.300000+7-3.165007-1 1.400000+7-3.471231-12631 3 2 8
9 1.500000+7-3.674350-1 1.600000+7-3.788282-1 1.700000+7-3.825634-12631 3 2 9
10 1.800000+7-3.799978-1 1.900000+7-3.725944-1 2.000000+7-3.617662-12631 3 2 10
11 2.200000+7-3.342054-1 2.400000+7-3.031296-1 2.600000+7-2.709398-12631 3 2 11
12 2.800000+7-2.388019-1 3.000000+7-2.077513-1 3.500000+7-1.398614-12631 3 2 12
13 4.000000+7-8.801994-2 4.500000+7-5.013441-2 5.000000+7-2.349373-22631 3 2 13
14 5.500000+7-5.480233-3 6.000000+7 6.189980-3 6.500000+7 1.332313-22631 3 2 14
15 7.000000+7 1.727774-2 7.500000+7 1.906833-2 8.000000+7 1.943711-22631 3 2 15
16 9.000000+7 1.783081-2 1.000000+8 1.499871-2 1.100000+8 1.211197-22631 3 2 16
17 1.200000+8 9.641120-3 1.300000+8 7.717424-3 1.400000+8 6.318436-32631 3 2 17
18 1.500000+8 5.367555-3 1.600000+8 4.771939-3 1.800000+8 4.312985-32631 3 2 18
19 2.000000+8 4.445323-3                  2631 3 2   19

```

---

### 4.4.4 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,  $\lambda$ , for the radioactive isotope. This constant is defined as follows:

$$\lambda = -\frac{N'(t)}{N(t)} \quad (4.6)$$

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:

$$N(t) = N(0) \exp(-t\lambda) \quad (4.7)$$

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:

$$A(t) = -N'(t) = \lambda N(t) \quad (4.8)$$

$$A(t) = \lambda N(0) \exp(-t\lambda) \quad (4.9)$$

#### 4.4.5 Saturation Activity

As a radioactive isotope is created by reactions between target atoms and projectiles (protons, neutrons, deuterons) it will begin to decay. The amount of the isotope will continue to increase until the decay rate equals the reaction rate for the creation of the isotope.

For a single isotope:

$$\omega = \frac{J}{Q} \cdot n_p \cdot \sigma \cdot 10^{-28} \delta t \frac{dN}{dt} = \omega - \lambda N, N(0) = N_0 \quad (4.10)$$

This equation may be solved using Laplace transforms (appendix ??) and this will be explored in more detail in section 6.2.1. The solution for a single isotope that has a starting amount  $N_0$ , a decay constant  $\lambda$  and a creation rate of  $\omega$  is given in eq. 4.11.

$$N(t) = \frac{\omega}{\lambda} + \left( N_0 - \frac{\omega}{\lambda} \right) \exp(-\lambda t) \quad (4.11)$$

The saturation activity may be computed as follows. As  $t$  approaches infinity the exponential term tends towards zero due to the negative decay constant. Replacing the exponential terms with zero and noting eq. 4.8 leaves a simple result to compute the saturation activity  $A_s$  (eq. 4.13). This is logical as, once saturated, the isotope will decay at the same rate as its source.

$$N_s = \frac{\omega}{\lambda} \\ A_s = \lambda N_s = \omega \quad (4.12)$$

The saturation time is trickier to compute. Mathematically the sample would never reach saturation, it would get ever closer as time elapsed. Rather than compute this time (which would be meaningless) it is more useful to compute the time at which the saturation reaches a fraction,  $k$ , of the maximum saturation amount  $N_s$ .

$$kN_s = \frac{\omega}{\lambda} + \left( N_0 - \frac{\omega}{\lambda} \right) \exp(-\lambda t) \\ t = \frac{-1}{\lambda} \ln \left[ \frac{k-1}{N_0 \frac{\lambda}{\omega} - 1} \right] \quad (4.13)$$

The equation is useful in calculating (near to) the maximum possible activity of a isotope being created in a proton beam or other similar process where radioactive isotopes are created and lost at a constant rate.

#### 4.4.6 Decay Constants: Joint Evaluated Fission and Fusion File (JEFF) 3.3 Data File

JEFF 3.3 is an evaluated data file[22], meaning it has been evaluated bu a relevant authority. The quality of the data may affect the health of the public and industry workers, so it must be evaluated. This particular file is managed by and available through the Nuclear Energy Agency (NEA).

It is a collection of many data files. Released in 2017, it also contains several files for incident gammas, protons, deuterons, tritons, helium-3 and alphas from the TENDL 2017 data file.

The files that will be important for this work are the ENDF-6 radioactive decay data files only. The nuclear reaction cross section data will come from Talys and various iterations of the TENDL libraries. The decay data held in the JEFF 3.3 file includes isotope data, masses, half lives, branching factors, continuous and discrete gamma data.

#### 4.4.7 Bateman Equation for Radioactive Decay

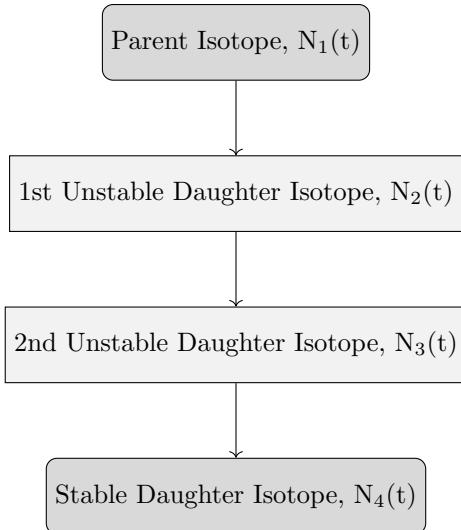


Figure 4.9: An example decay chain from an unstable parent isotope, through unstable daughter isotopes ending with a stable daughter isotope.

The English mathematician Harry Bateman derived an equation (eq. 4.14) to calculate the amount of each isotope in a decay chain, illustrated in Figure 4.9, at time t.

$$N_n(t) = \sum_{i=1}^{i=n} \left( \left( \prod_{j=i}^{j=n-1} \lambda_{(ij+1)} \right) \sum_{j=i}^{j=n} \left( \frac{N_{i0} \exp(-\lambda_j t)}{\prod_{p=i, p \neq j}^{p=n} (\lambda_p - \lambda_j)} \right) \right) \quad (4.14)$$

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) [22].

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.

## 4.5 Simulating Ion Irradiation with SRIM and TRIM

A package of ion transport codes, SRIM, is freely available to download and use to investigate the transport of ions through matter. It includes tools to calculate the stopping range of a given ion in a material as well as TRTransport In Matter, a code used to calculate the energy and position of an ion through a target.

TRIM does not take into account the structure of a target. It may be layered in the calculation perpendicular to the beam, but beyond that the target is treated as amorphous. It will give the same result for BCC iron as it would for FCC iron, providing the density remains the same. The density and whether or not the target layer

is a gas or solid are options that must be configured when setting up a calculation. Each ion history is tracked through the same target; the target never changes. TRIM is ignorant to the structure and therefore damage to the structure that would accumulate over time, and to any swelling and change in composition or density.

The interaction of the ion with the target material is split into two parts: nuclear loss and electronic loss. The ion is tracked through the target until either its energy drops below a set threshold (10eV) or it leaves the target (fig 4.10). The code then moves on to the next ion history.

In a projectile-nucleon interaction, the magic formula (eq. 4.15)[93] is used to determine the scattering angle  $\Theta$  of the ion and target nucleus, as well as the amount of energy transferred. The azimuthal angle is selected randomly by multiplying  $2\pi$  by a random float [0.0, 1.0].

$$\begin{aligned}
 \cos\left(\frac{\Theta}{2}\right) &= \frac{B + R_c + \Delta}{R_0 + R_C} \\
 B &= \frac{p}{a} \\
 R_0 &= \frac{r_0}{a} \\
 R_C &= \frac{\rho}{a} \\
 \Delta &= A \frac{R_0 - B}{1 + G} \\
 A &= 2\alpha\epsilon B^\beta \\
 G &= \gamma \left( (1 + A^2)^{\frac{1}{2}} - A \right)^{-1} \\
 \alpha &= 1 + C_1\epsilon^{-\frac{1}{2}} \\
 \beta &= \frac{C_2 + \epsilon^{\frac{1}{2}}}{C_3 + \epsilon^{\frac{1}{2}}} \\
 \gamma &= \frac{C_4 + \epsilon^{\frac{1}{2}}}{C_5 + \epsilon^{\frac{1}{2}}}
 \end{aligned} \tag{4.15}$$

where  $p$  is the impact parameter

where  $a$  is the screening length

where  $r_0$  is the distance of closest approach

where  $\epsilon$  is the reduced energy

The potential between the projectile and target is calculated at the distance of closest approach,  $r_0$ . It is a Coulomb type potential (eq. 4.16) than includes the universal screening function (eq. 4.17). This potential will appear again, although occasionally with differing parameters, in chapter 5.

$$V(R) = \frac{Z_1 Z_2 e^2}{a R} \Phi(R) \tag{4.16}$$

$$\Phi(R) = 0.1818 \exp(-3.2x) + 0.5099 \exp(-0.9423x) + 0.2802 \exp(-0.4028x) + 0.2817 \exp(-0.2016x) \tag{4.17}$$

The recoil nuclei are also followed through several generations until their energies fall below that set for either the surface binding energy or displacement energy (3-6eV and 15-30eV respectively)[93].

The free flight path between large collisions is calculated based on the projectile energy, type and the target material. The path length between collisions is randomly generated by taking the maximum free path and

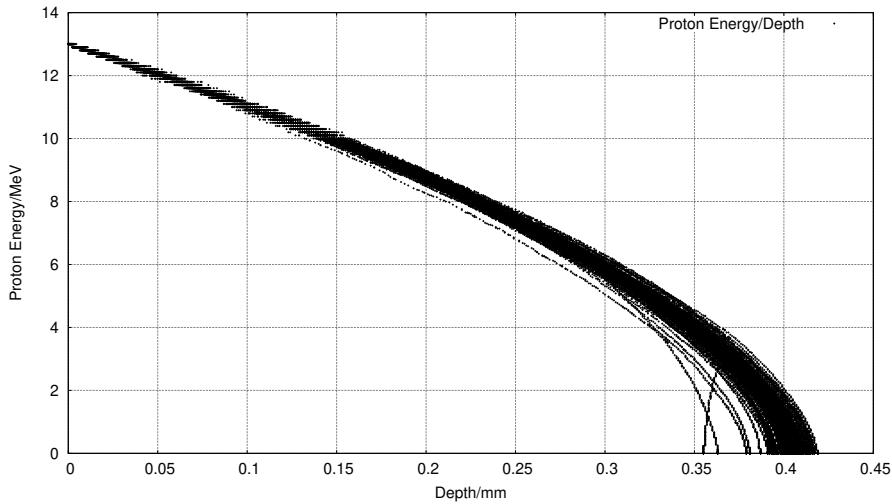


Figure 4.10: One hundred simulated 13 MeV proton energy loss curves in Fe simulated with SRIM [94]

multiplying by a random float [0.0, 1.0]. Smaller projectile-nucleus interactions are not individually calculated, but the average energy that would have been lost is calculated and applied.

The ion continuously loses energy to the electrons in the material. TRIM calculates this over the free flight path,  $L$  (eq. 4.18).

$$\Delta E_e = L N S_e(E) \quad (4.18)$$

where  $S_e(E)$  is the electronic stopping  
and  $N$  is the atomic density of the target

The target damage is a combination of the incoming ion, the primary knock on atoms and their damage cascades. The user has the option of running a full damage cascade that plots the entire cascade, or a quicker calculation where the cascade is treated as just a point. The total number of displacements is the sum of vacancies and replacement collisions, and the vacancies is sum of interstitials with the number of atoms that leave the target.

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 4.10 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 Figure 4.10.

# Chapter 5

## Background: Interatomic Potential Fitting

*In order to model Fe and Pd with Molecular Dynamics codes, an interatomic potential is needed. This will require experimental data and data generated by first principles calculations. The simplified model will consist of just Fe and Pd, but pure Fe does not take the FCC structure that it does when alloyed with Nickel in Austenitic Stainless Steel.*

*The properties of theoretical pure FCC Fe are estimated using Density Functional Theory that solves the many body Schrödinger Equation. While it is impossible to solve this equation exactly, with our current technology and knowledge, it is possible to solve approximately by making several approximations and by using the Hohenberg-Kohn theorem.*

*Many body potentials such as the FS and embedded-atom method are often used to model metals. The force matching method is used to fit the potentials to data. Optimization algorithms are split into global and local, with an example of a global being simulated annealing and a local being BFGS.*

### 5.1 Experiment, Modelling and Theory

#### 5.1.1 Introduction

We do not have a complete understanding of the world around us, so it may be that experiment is the “best” method of testing a material, as by its nature it will obey the rules of physics. As far as we know, these do not change from one place to another. A piece of steel used in an experiment is never going to have precisely the same composition and structure as similar steel use in a reactor, but it should be close enough.

The scanning electron microscope has a magnification of approximately 1 million times and they are able to generate a picture of the surface of a specimen using a beam of electrons. The transmission electron microscope has the ability to magnify a specimen further, with a resolution of less than 1 angstrom. The trade off is that it covers a smaller area of the specimen than a SEM and the sample must be prepared in thin slices.

The composition of an irradiated sample may be tested by a number of methods including neutron activation analysis and secondary-ion mass spectrometry. Scattering techniques, such as x-ray or neutron diffraction, can give an insight into the structure of a material, but they are statistical methods and do not give detailed measurements of individual atoms or defects.

There are draw backs to experiments, several of which are particular to this work. Even with our advances in technology, there are still errors introduced into any experiment. When dealing with smaller and smaller volumes of atoms quantum effects will also need to be considered, but this is perhaps a difficultly in our inability to visualise what happens on such small scales.

Radiation damage experiments, whether by neutrons or ions, will result in radioactive waste and the target must be safe to handle to be examined. The collision event happens up to a micron or more into the material. They occur on a very short timescale and within small volumes of space at random points along the path of the projectile, so the events would be very hard or impossible to track as they occur with current technology.

Current theories have been developed and modified over hundreds of years, and in the last 100 years Quantum Mechanics has been introduced to better describe physics on a small scale. It is a well proven theory that has been very successful. Unfortunately, using the theory exactly for a large ensemble of atoms and electrons is impossible with our current level of technology.

Computer models use theory, and a number of approximations, to simulated reality. These models may need to be fine tuned, to better fit both the theory and experiment, but they allow us to do what is not possible with either experiment or theory. It is possible to simulate an individual damage cascade and watch it in fine detail spatially and temporally. As computer power increases, the models used become more complex and the simulation boxes and time spans grow.

This chapter discusses the background required to link the more accurate DFT to MD by deriving interatomic potentials based on experiment and calculated data. By doing this, the concentrations at the grain boundary for an iron-palladium alloy may be modelled following irradiation damage.

### 5.1.2 Simulating Materials on a Variety of Scales in Time and Space

Taking a 1 micron grain in a steel as an example, it would contain tens of billions of atoms. There are a wide range of modelling programs available (fig. 5.1).

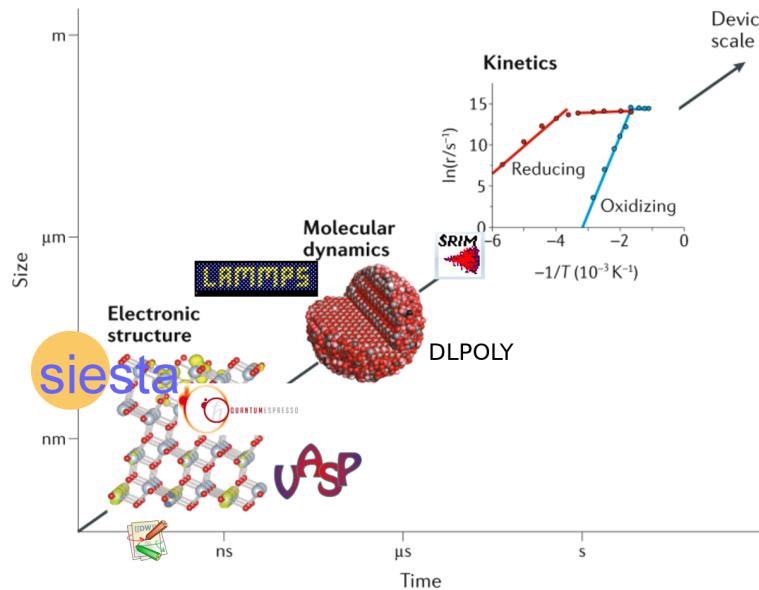


Figure 5.1: Time and Size Scales for Computer Packages [95]

The properties of metals may be approximated using small crystals, and this is ideally suited to DFT calculations. Simulations of small ensembles of atoms over time are now also possible using DFT molecular dynamics. Larger simulations containing thousands to millions of atoms, that are able to simulate single damage cascades are suitable for MD simulations.

## 5.2 Properties of Crystals

### 5.2.1 Introduction

There are seven crystal classes (table. 5.1) and 14 Bravias lattices (section 5.8.6), although this work is only concerned with cubic and orthorhombic crystals.

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

Table 5.1: Seven Crystal Classes

Metals are rarely single crystals, but are made from many crystals separated by grain boundaries. However, the knowledge of the microscopic crystals translates well to the properties of the metals on a macroscopic scale.

This work does not attempt to calculate properties based on a collection of many crystals, but on a single crystal with less than a thousand atoms. The crystals are then assumed to be infinite in size, due to periodic boundary conditions.

### 5.2.2 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.

#### 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 and several example values are given in table 5.2. 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.

$$B_0 = -V \frac{\partial P}{\partial V} \quad (5.1)$$

$$B_0 = V \frac{\partial^2 E}{\partial V^2} \quad (5.2)$$

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

Table 5.2: Example bulk modulii for three metals

**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[96]. Muraghan derived a new equation to improve upon formulae developed in the 1930's, using compression data from high pressure experiments.

$$P(V) = \frac{B_0}{B'_0} \left( \left( \frac{V_0}{V} \right)^{B'_0} - 1 \right) \quad (5.3)$$

As pressure is the negative derivative of the internal energy of the system with respect to change in volume,  $p = -(\partial E/dV)$ , and the equation can be integrated and written in terms of the energy, volume, bulk modulus and its derivative[97].

$$E(V) = E_0 + \frac{B_0 V}{B'_0} \left[ \left( \frac{V_0}{V} \right)^{B'_0} \frac{1}{B'_0 - 1} + 1 \right] - \frac{B_0 V_0}{B'_0 - 1} \quad (5.4)$$

**Birch-Murnaghan Equation of State**

Several years after Murnaghan's equation, Birch developed the equation of state further upon the experimental data provided by work from Bridgman in high pressure physics. For cubic symmetry, the description of free energy now includes third order terms in the strain components[98].

$$P(V) = \frac{3B_0}{2} \left[ \left( \frac{V_0}{V} \right)^{\frac{7}{3}} - \left( \frac{V_0}{V} \right)^{\frac{5}{3}} \right] \left[ 1 + \frac{3}{4}(B'_0 - 4) \left( \left( \frac{V_0}{V} \right)^{\frac{2}{3}} - 1 \right) \right] \quad (5.5)$$

The energy-volume relationship may again be constructed[97].

$$E(V) = E_0 + \frac{9V_0 B_0}{16} \left[ \left[ \left( \frac{V_0}{V} \right)^{\frac{2}{3}} - 1 \right]^3 B'_0 + \left[ \left( \frac{V_0}{V} \right)^{\frac{2}{3}} - 1 \right]^2 \left[ 6 - 4 \left( \frac{V_0}{V} \right)^{\frac{2}{3}} \right] \right] \quad (5.6)$$

In order to fit the Birch-Murnaghan, the first step is to fit a second order polynomial to the energy-volume data. This may be achieved using least-squares fitting with a vandermonde matrix. The coefficients from this polynomial may then be used to calculate reasonable value for  $E_0$ ,  $V_0$  and  $B_0$ ; sane starting values of  $B'_0$  are between 1 and 10, and the code takes a starting value of 2[99].

$$\begin{aligned}
E(V) &= c_0 + c_1 V + c_2 V^2 \\
V_0 &= -\frac{c_1}{2c_2} \\
E_0 &= c_2 \times V_0^2 + c_1 V_0 + c_0 \\
B_0 &= 2c_2 V_0 \\
B'_0 &= 2
\end{aligned} \tag{5.7}$$

Newton Gauss (eq. 5.8) may then be used to attempt to minimise the fit further for  $E_0$ ,  $V_0$  and  $B_0$  whilst trying values of  $B'_0 \in 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0$ . The parameters with the lowest residual square sum are returned.

$$[J^T J] P = J^T R \tag{5.8}$$

Mehl et al used a version of the Birch equation of state[98] and suggest a value for  $B'_0$  between 3 and 5[100] and limit the value of N in eq. 5.9 to either 3 or 4 for the best data fit.

$$E(V) = E_0 + \frac{9}{8} B_0 V_0 \left[ \left( \frac{V_0}{V} \right)^{\frac{2}{3}} - 1 \right]^2 + \frac{9}{16} B_0 (B'_0 - 4) V_0 \times \left[ \left( \frac{V_0}{V} \right)^{\frac{2}{3}} - 1 \right]^3 + \text{sum}_{n=4}^N \gamma_n \left[ \left( \frac{V_0}{V} \right)^{\frac{2}{3}} - 1 \right]^n \tag{5.9}$$

In generating the data for a Birch or Birch-Murnaghan fit the atom configuration is relaxed. A small strain  $[\sigma, \sigma, \sigma, 0.0, 0.0, 0.0]$  is applied to decrease and increase the volume slightly about the relaxed volume.

### Rose-Vinet Equation of State

$$E(a) = E_0 + \left( 1 + -\alpha \left( \frac{a}{a_0 - 1} \right) \right) \exp \left( -\alpha \left( \frac{a}{a_0 - 1} \right) \right) \alpha = \sqrt{\frac{9\Omega B_0}{-E_0}} \tag{5.10}$$

More recently a Universal form, eq 5.10, was proposed by Rose, Vinet et al[101]. With an exponential cut off, this fits from the relaxed lattice parameter to much wider separations with a good fit for simple and transition metals, including K, Mo, Cu (fig. 5.2).

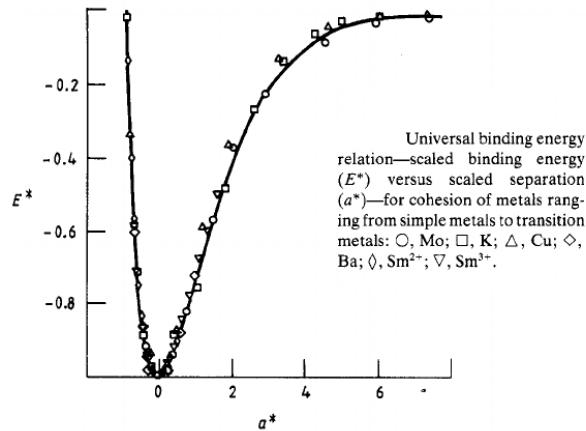


Figure 5.2: Rose-Vinet Equation of State

This has been used as a tool to fit interatomic potentials in work by Jelinek et al[102] and Bonny et al[103]. The version of the equation in this work is that used by Bonny et al and their fitting code, which will be discussed in more detail in section 5.7.

### 5.2.3 Voigt Notation

Where a tensor is symmetric, Voigt notation is used to simplify how the tensor is written. It reduces a second order tensor such as stress or strain, represented by a 3x3 matrix, to a 6 row matrix. It also reduces a fourth order tensor, such as the compliance or stiffness tensor (represented by a 3x3x3x3 matrix), to a 6x6 matrix.

$$\vec{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} A_{11} \\ A_{22} \\ A_{33} \\ A_{23} \\ A_{13} \\ A_{12} \end{bmatrix} \text{ if } \vec{A} \text{ is symmetric} \quad (5.11)$$

### 5.2.4 Elastic Constants

Stress and strain are both related by either the stiffness tensor  $C$ , filled with the elastic constants, the compliance tensor  $S$  and this is the inverse of the stiffness tensor. Strain is a measure of the deformation of a material and by definition has no units. In one dimension  $\epsilon = \frac{\delta l}{l_0}$  where  $l_0$  is the unstrained length and  $\delta l$  is the change in length. The Generalized Hooke's law relating these tensors allows the computation of the Cauchy stress if the elastic constants are known and an input strain is provided.

$$\vec{A} = \begin{bmatrix} \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{21} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{31} & \epsilon_{32} & \epsilon_{33} \end{bmatrix} = \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{23} \\ \epsilon_{13} \\ \epsilon_{12} \end{bmatrix} \text{ where } \epsilon_{ij} \equiv \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (5.12)$$

$$\vec{\sigma}_{ij} = \vec{C}_{ijkl} \vec{\epsilon}_{kl} \quad (5.13)$$

$$\vec{\epsilon}_{ij} = \vec{S}_{ijkl} \vec{\sigma}_{kl} \quad (5.14)$$

$$\epsilon_{ij} = \begin{bmatrix} \epsilon_1 = \epsilon_{11} = \epsilon_{11} \\ \epsilon_2 = \epsilon_{22} = \epsilon_{22} \\ \epsilon_3 = \epsilon_{33} = \epsilon_{33} \\ \epsilon_4 = \epsilon_{23} = \epsilon_{32} \\ \epsilon_5 = \epsilon_{13} = \epsilon_{31} \\ \epsilon_6 = \epsilon_{12} = \epsilon_{21} \end{bmatrix} \quad (5.15)$$

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix} \quad (5.16)$$

The number of independent elastic constants changes depending on the symmetry of the crystal. The tensors in Voigt notation are given for cubic, orthorhombic/orthotropic, monoclinic and triclinic in appendix ??.

### 5.2.5 Calculating Elastic Constants for a Cubic Crystal

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. Due to symmetry, a cubic crystal has only three elastic independent constants; in Voigt notation these are  $C_{11}$ ,  $C_{12}$  and  $C_{44}$ .

Applying two strains to a cubic crystal [104] coupled with the calculation of the Bulk Modulus, as already discussed, allows the three independent elastic constants to be calculated.

First, the bulk modulus may be calculated either by finding the second derivative of the energy wrt volume at the relaxed volume,  $B(V) = -VP'(V) = VE''(V)$ [104], or by fitting the Birch-Murnaghan equation of state.

$$\epsilon_{(C11-C22)} = \begin{bmatrix} \delta & 0 & 0 \\ 0 & -\delta & 0 \\ 0 & 0 & \delta^2/(1-\delta^2) \end{bmatrix} \quad (5.17)$$

Second, a volume conserving orthorhombic strain (eq. 5.17) may be applied to the crystal [104]. The relation between the relaxed energy and that under strain is symmetric about  $E(0)$  and is given by  $E(\delta) = E(-\delta) = E(0) + (C_{11} - C_{12})V\delta^2 + O[\delta^4]$ [104].  $V$  is the volume of the relaxed unit cell and  $E(0)$  is the minimum energy. By fitting a polynomial to the energy-strain data, the coefficient for the quadratic term is equal to  $(C_{11} - C_{12})V$ .

$$\epsilon_{(C44)} = \begin{bmatrix} 0 & \frac{\delta}{2} & 0 \\ \frac{\delta}{2} & 0 & 0 \\ 0 & 0 & \delta^2/(4-\delta^2) \end{bmatrix} \quad (5.18)$$

Third, a volume conserving monoclinic strain (eq. 5.18) is applied and the relation between the relaxed energy, strained energy and the elastic constant  $C_{44}$  is  $E(\delta) = E(-\delta) = E(0) + \frac{1}{2}C_{44}V\delta^2 + O[\delta^4]$ . Similarly, fitting a polynomial to the strain-energy data points will calculate the value of  $C_{44}$ .

Finally, the relation between the bulk modulus,  $C_{11}$  and  $C_{12}$ ,  $B_0 = (C_{11} + 2C_{12})/3$  allows the computation of the individual constants eq. 5.19 (this relationship is only for cubic crystals).

$$\begin{aligned} C_{ortho} &= C_{11} - C_{12} \\ C_{11} &= \frac{3B_0 + 2C_{ortho}}{3} \\ C_{12} &= \frac{3B_0 - C_{ortho}}{3} \end{aligned} \quad (5.19)$$

### 5.2.6 Calculating Elastic Constants for Orthorhombic Crystal

The DFT work here includes Pd and Fe. 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 z-axis, the crystal becomes slightly tetragonal with one side being 5% larger than the other two.

$$C_{ij} = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{bmatrix} \quad (3 \text{ independent values}) \quad (5.20)$$

$$C_{ij} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \quad (9 \text{ independent values}) \quad (5.21)$$

Once the relaxed crystal basis vectors and atom positions have been determined for the orthorhombic crystal, nine strains are applied to the crystal [105] in order to calculate the nine independent elastic constants.

As a strain is applied to a crystal its energy can change. The volume may also change, although it may be held constant. The relationship between the energy and the strain  $\sigma$  may be represented as a Taylor expansion of the internal energy in powers of the strain tensor[105]. The equation used by Ravindran et al is in a slightly different notation to that used by Mehl[106]. It includes a constant  $xi$  to account for the symmetry of  $\sigma$  and the use of Voigt notation.

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

where  $\sigma$  is the strain  
and  $\tau$  is the stress  
if the index  $k$  of  $\xi$  is  $\in 1, 2, 3$  then  $\xi_k = 1$   
if the index  $k$  of  $\xi$  is  $\in 4, 5, 6$  then  $\xi_k = 2$

(5.22)

The first three strains applied to the orthorhombic crystal are non-volume conserving. They strain the crystal along each axis and allow the calculation of  $C_{11}$ ,  $C_{22}$  and  $C_{33}$ .

$$D_1 = \begin{bmatrix} 1 + \delta & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.23) \quad E(V, \sigma) = E(V_0, 0) + V_0 \left( \tau_1 \sigma + \frac{c_{11}}{2} \sigma^2 \right) \quad (5.24)$$

$$D_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 + \delta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.25) \quad E(V, \sigma) = E(V_0, 0) + V_0 \left( \tau_2 \sigma + \frac{c_{22}}{2} \sigma^2 \right) \quad (5.26)$$

$$D_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 + \delta \end{bmatrix} \quad (5.27) \quad E(V, \sigma) = E(V_0, 0) + V_0 \left( \tau_3 \sigma + \frac{c_{33}}{2} \sigma^2 \right) \quad (5.28)$$

Volume conserving monoclinic distortions (eq. 5.29, eq. 5.31, eq. 5.33) are used to calculate the  $C_{44}$ ,  $C_{55}$  and  $C_{66}$  elastic constants (eq. 5.30, eq. 5.32, eq. 5.34).

$$D_4 = \begin{bmatrix} \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & 0 & 0 \\ 0 & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & \frac{\sigma}{(1-\sigma^2)^{\frac{1}{3}}} \\ 0 & \frac{\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} \end{bmatrix} \quad E(V, \sigma) = E(V_0, 0) + V_0 \left( 2\tau_4 \sigma + 2 \frac{c_{44}}{2} \sigma^2 \right) \quad (5.30)$$
(5.29)

$$D_5 = \begin{bmatrix} \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & 0 & \frac{\sigma}{(1-\sigma^2)^{\frac{1}{3}}} \\ 0 & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & 0 \\ \frac{\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & 0 & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} \end{bmatrix} \quad E(V, \sigma) = E(V_0, 0) + V_0 \left( 2\tau_5 \sigma + 2 \frac{c_{55}}{2} \sigma^2 \right) \quad (5.32)$$
(5.31)

$$D_6 = \begin{bmatrix} \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & \frac{\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & 0 \\ \frac{\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & 0 \\ 0 & 0 & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} \end{bmatrix} \quad E(V, \sigma) = E(V_0, 0) + V_0 \left( 2\tau_6 \sigma + 2 \frac{c_{66}}{2} \sigma^2 \right) \quad (5.34)$$
(5.33)

Finally, three orthorhombic strains (eq. 5.35, eq. 5.37, eq. 5.39) that conserve the volume, by altering the strain along each axis, are used to calculate the  $C_{12}$ ,  $C_{13}$  and  $C_{23}$  elastic constants (eq. 5.36, eq. 5.38, eq. 5.40).

$$D_7 = \begin{bmatrix} \frac{1+\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & 0 & 0 \\ 0 & \frac{1-\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & 0 \\ 0 & 0 & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} \end{bmatrix} \quad (5.35)$$

$$E(V, \sigma) = E(V_0, 0) + V_0 \left( (\tau_1 - \tau 2\sigma + \frac{1}{2}(c_{11} + c_{22} - 2c_{12})\sigma^2) \right) \quad (5.36)$$

$$D_8 = \begin{bmatrix} \frac{1+\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & 0 & 0 \\ 0 & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & 0 \\ 0 & 0 & \frac{1-\sigma}{(1-\sigma^2)^{\frac{1}{3}}} \end{bmatrix} \quad (5.37)$$

$$E(V, \sigma) = E(V_0, 0) + V_0 \left( (\tau_1 - \tau 3\sigma + \frac{1}{2}(c_{11} + c_{33} - 2c_{13})\sigma^2) \right) \quad (5.38)$$

$$D_9 = \begin{bmatrix} \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & 0 & 0 \\ 0 & \frac{1+\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & 0 \\ 0 & 0 & \frac{1-\sigma}{(1-\sigma^2)^{\frac{1}{3}}} \end{bmatrix} \quad (5.39)$$

$$E(V, \sigma) = E(V_0, 0) + V_0 \left( (\tau_2 - \tau 3\sigma + \frac{1}{2}(c_{22} + c_{33} - 2c_{23})\sigma^2) \right) \quad (5.40)$$

### 5.2.7 Stability Conditions

Whilst the elastic constants may be computed using first-principles calculations, there are a series of assumptions and approximations made in order to solve first-principles calculations in a reasonable amount of time. There must be a sanity check, as it is no use to compute the elastic constants from DFT to find the crystals are unstable.

For a cubic crystal with three independent elastic constants and zero stress, the Born elastic stability criteria apply[107]. The bulk modulus should be positive  $(C_{11} + 2C_{12})/3 > 0$ , the tetragonal shear should be positive  $C_{11} - C_{12} > 0$  and the shear modulus should be positive  $c_{44} > 0$ .

For an orthorhombic crystal, there are additional conditions given there are nine independent elastic constants (eq. 5.41)[108].

$$\begin{aligned} C_{11} &> 0, C_{22} > 0, C_{33} > 0, C_{44} > 0, C_{55} > 0, C_{66} > 0 \\ (C_{11} + C_{22} - 2C_{12}) &> 0, (C_{11} + C_{33} - 2C_{13}) > 0, (C_{22} + C_{33} - 2C_{23}) > 0 \\ (C_{11} + C_{22} + C_{33} + 2C_{12} + 2C_{13} + 2C_{23}) &> 0 \end{aligned} \quad (5.41)$$

### 5.2.8 Computing Values from Elastic Constants

The bulk modulus, as noted earlier, is a measure of the effect of strain on stress. While it may be calculated by taking the second derivative of energy with respect to volume, or by fitting an equation of state, it may also be calculated using the elastic constants of a material or the compliance constants.

For cubic crystals the elastic constants may be used to calculate the bulk modulus, tetragonal shear, shear modulus and Cauchy pressure (eq. 5.42).

$$\begin{aligned}
 B_0 &= \frac{C_{11} + 2C_{12}}{3} = \frac{5C_{11}}{9} \text{ Bulk Modulus} \\
 C' &= \frac{C_{11} - C_{12}}{2} \text{ Tetragonal Shear} \\
 &\quad C_{44} \text{ Shear Modulus} \\
 p_c &= C_{12} - C_{44} \text{ Cauchy Pressure}
 \end{aligned} \tag{5.42}$$

For an orthorhombic crystal, the Voigt bulk modulus (eq. 5.43) and Reuss bulk modulus (eq. 5.44) may be calculated from the stiffness and compliance matrix respectively. These two values are then averaged to give the bulk modulus (eq. 5.45)[105].

$$B_V = \frac{1}{9} (C_{11} + C_{22} + C_{33} + 2(C_{12} + C_{13} + C_{23})) \tag{5.43}$$

$$B_R = (S_{11} + S_{22} + S_{33} + 2(S_{12} + S_{13} + S_{23}))^{-1} \tag{5.44}$$

$$B = 0.5(B_V + B_R) \tag{5.45}$$

Calculation by this method is particularly useful for this work, where the FCC iron crystal is orthorhombic, and not cubic (section 9.4.2).

Elastic constants are also used to calculate the shear modulus  $G$ . Where the crystal is cubic, there are several elastic constants that can be used (eq. 5.46)[105]. Both the Voigt (eq. 5.47) and Reuss (eq. 5.47) values are calculated and averaged to give the value of  $G$  (eq. 5.49).

$$G = C_{44} = 0.5(C_{11} - C_{12}) = \frac{C_{11}}{3} \tag{5.46}$$

$$G_V = \frac{1}{15} (C_{11} + C_{22} + C_{33} - C_{12} - C_{13} - C_{23}) + \frac{1}{5} (C_{44} + C_{44} + C_{55}) \tag{5.47}$$

$$G_R = \frac{15}{4(S_{11} + S_{22} + S_{33}) - 4(S_{12} + S_{23} + S_{23}) + 3(S_{44} + S_{55} + S_{66})} \tag{5.48}$$

$$G = 0.5(G_V + G_R) \tag{5.49}$$

The Young's modulus  $E$  as a scalar is a measure of how easily the length an isotropic material is changed under tension or compression, and the computed elastic constants for a cubic crystal may be used to calculate  $E$  (eq. 5.50).

$$E = \frac{9B_0G}{3B_0 + G} \text{ where } B_0 \text{ is the bulk modulus and } G \text{ is the shear modulus} \quad (5.50)$$

The Poisson ratio is a measure of how much a material changes in the direction perpendicular to a strain applied transversely.

$$\nu = -\frac{d\epsilon_{perp}}{d\epsilon_{transverse}} \quad (5.51)$$

The bulk modulus may be calculated from the equation of state or elastic constants, and the shear modulus may be calculated from the elastic constants. In turn, the Poisson ratio is calculated from these.

$$\nu = \frac{3B_0 - 2G}{2(3B + G)} \quad (5.52)$$

The ratio may be used as another data point in order to fit a potential, or at the very least as a check to compare to the known value once a potential has been derived.

### 5.2.9 Correlation of Melting Temperature in Metals with Elastic Constants

There is a correlation between the melting temperature of metals and their elastic constants detailed in eq. 5.53 and fig. 5.3) [109].

$$T \approx 598.0 + 6.66 \times (C_{11} + C_{22} + C_{33}) - 0.003 \times (C_{11} + C_{22} + C_{33}) \quad (5.53)$$

Where the temperature is in K, and the elastic constants are in GPa

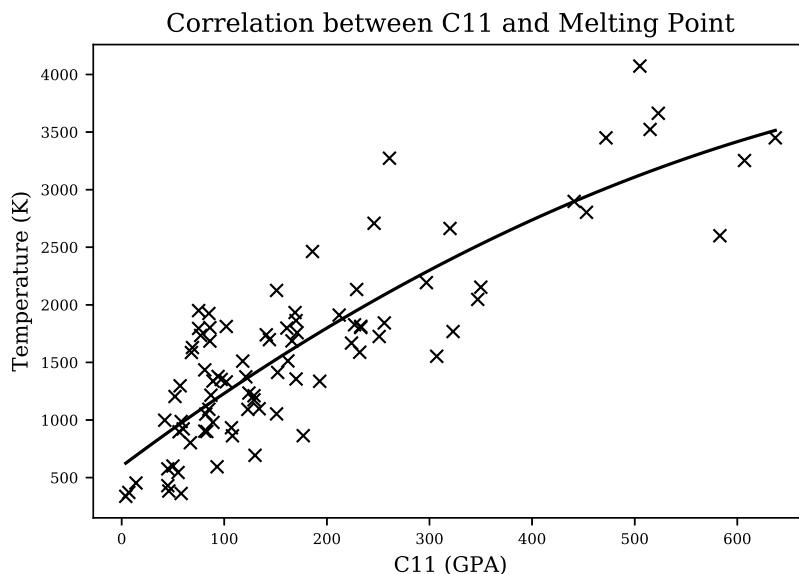


Figure 5.3: Correlation between  $C_{11}$  and temperature

The correlation values between the temperature and  $C_{11}$  are 0.848 and 0.780 for the Pearsons and Spearmans correlation respectively. An accurate temperature will not be possible to predict, but it will act as a further sanity check within the computer codes developed in the results section of this work.

## 5.3 Interatomic Potentials and Choice of Function

### 5.3.1 Introduction

An interatomic potential, as used in this work and Molecular Dynamics computer codes, is a function or a set of functions that describe the energy and force between atoms. Simpler functions represent the potential energy between pairs of atoms only, but more complicated functions have been used in molecular dynamics since the 1980s that also attempt to represent the many body nature of materials, which applies in particular to metals.

### 5.3.2 Pair Potentials

A pair potential only considers individual pairs of nearby atoms, one pair at a time, and does not consider the effect of any other nearby atoms. Where an alloy is being modelled, there will be a pair potential function for each element and element combination; 1 for a single element, 3 for a two element alloy, 6 for a three element alloy and so on.

#### Lennard-Jones

The Lennard-Jones potential was proposed in the 1920s and has both an attractive term and a repulsive term; the  $(r_m/r)^{12}$  becomes much larger than the  $(r_m/r)^6$  term at close distances, and this mimics the coulomb repulsion as two atoms are pushed closer together. At larger separations, the attractive term dominates.

$$V(r) = e \left( \left( \frac{r_m}{r} \right)^{12} - 2 \left( \frac{r_m}{r} \right)^6 \right) \quad (5.54)$$

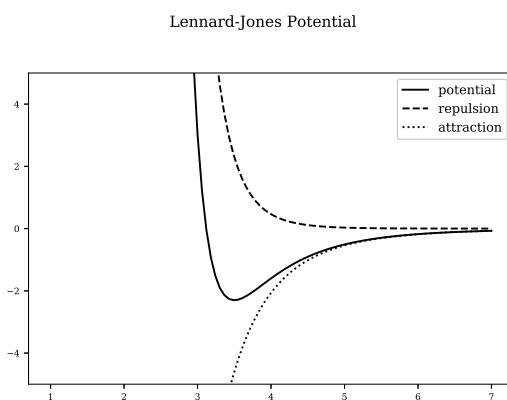


Figure 5.4: Lennard-Jones Potential

## Morse Potential

The Morse potential was proposed five years after the Lennard-Jones potential. It also has an attractive and repulsive term, but it uses the exponential function rather than 6th and 12th powers.

$$V(r) = \exp(-2a(r - re)) - 2 \exp(-a(r - re)) \quad (5.55)$$

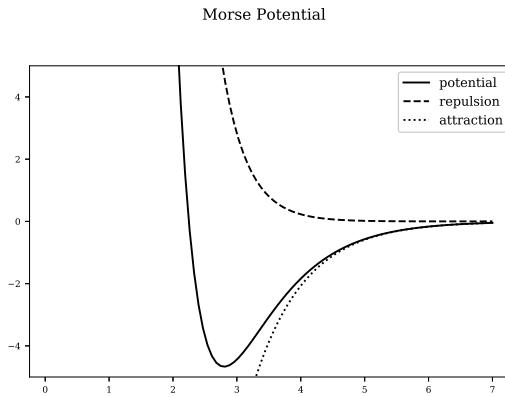


Figure 5.5: Morse Potential

## Buckingham Potential

The Buckingham Potential consists of a repulsive and attractive part. As the separation decreases, the attractive term dominates and the overall function tends towards negative infinity. This shouldn't be problematic when gasses or solids are being modelled alone, but when collisions and damage cascades are being modelled, the separation between atoms may be much smaller than that in a typical simulation, ending with these atoms being pulled together (eq. 5.56 and fig. 5.6).

$$V(r) = A \exp(-Br) - \frac{C}{r^6} \quad (5.56)$$

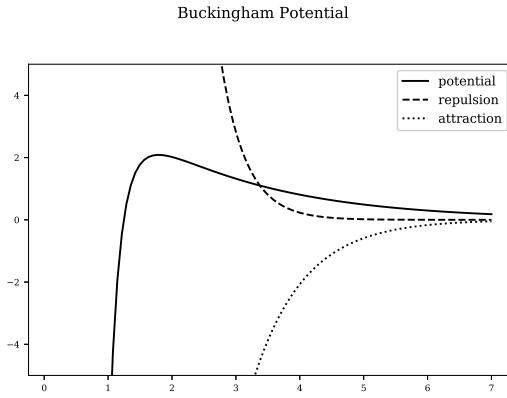


Figure 5.6: Buckingham Potential

### Ziegler-Biersack-Littmark Universal Potential 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 potential, between an atom of charge  $Z_i$  and  $Z_j$  is set out in the SRIM manual[93].

$$V_{zbl}(r_{ij}, Z_i, Z_j) = \frac{1}{4\pi\epsilon_0} \frac{Z_i Z_j}{r_{ij}} \phi\left(\frac{r_{ij}}{a_{ij}}\right) \quad (5.57)$$

where  $\epsilon_0 = 8.85419 \times 10^{-12}$

The parameters of the Universal screening potential function are as follows:

$$\phi(x) = 0.181e^{-3.2x} + 0.5099e^{-0.9423x} + 0.2802e^{-0.4029x} + 0.02817e^{-0.2016x} \quad (5.58)$$

where  $a_{ij} = \frac{0.8854a_0}{Z_i^{0.23} + Z_j^{0.23}}$

and  $a_0 = 0.529$  angstrom

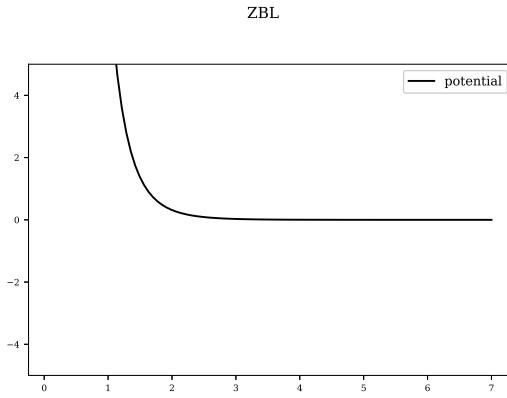


Figure 5.7: ZBL universal screening potential

### 5.3.3 Many Body Potentials

This work is focused on metals, and two popular, and closely related potentials, will be discussed in this section.

#### Finnis-Sinclair

The Finnis-Sinclair potential was published in 1984[110] 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.

The tight-binding that this potential is based on is represented by the functional: square root of the density. The potential for a single element model consists of a pair function, a density function and a tight-binding functional.

$$\begin{aligned} u^{A,A} &= \sum_{i \neq j}^N (r) v^{A,A}(r_{ij}) \\ \rho_i^A(r) &= \sum_{j,j \neq i}^N \phi(r_{ij}) \\ F^A &= \sum_i^N \rho_i \end{aligned} \tag{5.59}$$

#### Embedded-atom method

The embedded-atom method was also invented in the 1980s by Daw and Baskes. It was developed with metals in mind, and in many ways is a more flexible variant of the Finnis-Sinclair potential. It too has a pair and density function, but the functional of the EAM potential is not restricted to a square root.

$$\begin{aligned} U_{EAM} &= \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i}^N V_{ij}(r_{ij}) + \sum_{i=1}^N F[\rho_i] \\ \text{where } \rho_i &= \sum_{j=i, j \neq i}^N \rho_{ij}(r_{ij}) \end{aligned} \tag{5.60}$$

Elastic properties were not reliably calculated from pair functions alone[111], but the advent of Finnis-Sinclair and EAM type potentials changed this.

$$A_{ij} + F'(\vec{\rho})V_{ij} = 0$$

where

$$\begin{aligned} A_{ij} &= \frac{1}{2} \sum_m \phi'_m a_i^m a_j^m / a^m \\ V_{ij} &= \frac{1}{2} \sum_m \rho'_m a_i^m a_j^m / a^m \end{aligned} \quad (5.61)$$

For homonuclear cubic crystals, the lattice constant may be calculated by the equilibrium condition in eq 5.61. The three independent elastic constants,  $C_{11}, C_{12}, C_{44}$ , may also be calculated at equilibrium from eq 5.62.

$$C_{11} = (B_{11} + F'(\vec{\rho})W_{11} + F''(\vec{\rho})(V_{11})^2) / \Omega_0$$

$$C_{12} = (B_{12} + F'(\vec{\rho})W_{12} + F''(\vec{\rho})(V_{11})^2) / \Omega_0$$

$$C_{44} = (B_{12} + F'(\vec{\rho})W_{12}) / \Omega_0$$

where

$$B_{ijkl} = \frac{1}{2} \sum_m (\phi''_m - \phi'_m/a^m) a_i^m a_j^m a_k^m a_l^m / (a^m)^2 \quad (5.62)$$

$$W_{ijkl} = \frac{1}{2} \sum_m (\rho''_m - \rho'_m/a^m) a_i^m a_j^m a_k^m a_l^m / (a^m)^2$$

where

$$\phi''_m = (d^2\phi(r)/dr^2)_{r=a^m},$$

$$\rho''_m = (d^2\rho(r)/dr^2)_{r=a^m}.$$

There are notable consequences in removing either the pair function or embedding functional. If  $V_{ij}(r_{ij})$  is removed,  $F'[\vec{\rho}] = 0$  and this gives a shear modulus of 0 ( $C_{11} = C_{12}, C_{44} = 0$ )[111]. If  $F[\vec{\rho}]$  is removed the Cauchy discrepancy becomes zero ( $C_{12} = C_{44}$ )[111]. Both of these situations are generally untrue, and both functions are required for a good potential for a metal.

### Two band embedded-atom method

There are several variations of the potential, and one of particular interest to us is the two band embedded-atom method that has two electron density and embedding energy terms. This formalism was originally developed to model Caesium[112], and the transition of electrons between S and D bands under pressure, but it has been modified to apply to alloys.

Caesium changes its structure as pressure is applied. The first change is from BCC to a more compact FCC structure, but it then compresses further as electrons move from the S-band to the more compact D-band, reducing the size of the atom.

The bond energy may be described with eq. 5.63 where  $N_1$  and  $N_2$  are the capacities of each band,  $n_{i1}$  and  $n_{i2}$  are the electrons in each band of the  $i^{\text{th}}$  atom.  $E_{\text{prom}}$  is the energy required to promote an electron from band 1 to band 2.

$$U_{bond} = \sum_i \frac{W_{i1}}{2N_1} n_{i1}(n_{i1} - N_1) + \sum_i \frac{W_{i2}}{2N_2} n_{i2}(n_{i2} - N_2) + E_{prom} \quad (5.63)$$

A Finnis-Sinclair type EAM potential was used by Ackland and Reed in this work for both bands. The parameters fitted in Ackland and Reed's work are listed in table 5.3.

$$\begin{aligned} & \text{Pair} \\ V_s(r_{ij}) &= \sum_i \frac{A_s}{r_{ij}^{12}} \\ V_d(r_{ij}) &= \sum_i \frac{A_d}{r_{ij}^{12}} \end{aligned} \quad (5.64)$$

$$\rho_s(r_{ij}) = \begin{cases} C_s(d_s - r_{ij})^3 & r_{ij} < d_s \\ 0 & r_{ij} > d_s \end{cases} \quad \rho_d(r_{ij}) = \begin{cases} C_d(d_d - r_{ij})^3 & r_{ij} < d_d \\ 0 & r_{ij} > d_d \end{cases} \quad (5.65)$$

$$\text{Embedding} \quad (5.66)$$

Parameter	Value
$C_s$	$0.05617 eV^2 \text{angs}^{-3}$
$C_d$	$0.1681 eV^2 \text{angs}^{-3}$
$d_s$	$9.5097 \text{angs}$
$d_d$	$6.9189 \text{angs}$
$A_s$	$2.417 \times 10^7 \text{angs}^{12}$
$A_d$	$3.7668 \times 10^6 \text{angs}^{12}$

Table 5.3: Caesium 2BEAM parameters

The implementation of this two-band potential predicted the transformation of Caesium. The I phase, at ambient pressure, is BCC Caesium with a primitive cell volume of 115.9 cubic angstrom per atom. As more pressure is added, the optimum structure changes from BCC to FCC, and this results in a smaller volume per atom of 67.5 cubic angstrom. Finally Caesium undergoes an isostructural transformation, and the potential predicts a volume of 48.7 cubic angstrom per atom.

There is a transition pressure of 4.3 GPa between the phases II and III. The potentials developed by Ackland and Reed were “in good agreement with experiment”[112].

An alloy version of the two-band model was developed by Olsson et al. to investigate the  $\alpha$ -prime phase formation in Fe-Cr[113]. Fe-Cr alloys form ferromagnetic alloys with concentrations of up to 10% chromium at  $750^\circ\text{C}$ . As the concentration of chromium in the alloy increases, the alloy begins to decompose into iron rich and chromium rich precipitates, and this decomposition is accelerated under irradiation.

The two-band method for Caesium was extended to an Fe-Cr alloy in order to describe the heat of mixing in the alloy.

$$U_{EAM} = \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i}^N V_{ij}(r_{ij}) + \sum_{i=1}^N F_D[\rho_{d,i}] + \sum_{i=1}^N F_S[\rho_{s,i}] \quad (5.67)$$

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

The embedding functional (eq. 5.71) was in the form of several functionals used in papers by Ackland and Mendelev, and an extension to the standard Finnis-Sinclair.

$$F_{band}(\rho_{band}) = A_1^{band} \sqrt{\rho_{band}} + A_2^{band} \rho_{band}^2 + A_3^{band} \rho_{band}^4 \quad (5.68)$$

The choice of function for both the pair and d-band density functions was a cubic spline, and a 4s Slater type function was used for the s-band alloy density. The Fe-Fe and Cr-Cr s-band density functions were zero functions. Where the alloy has high concentrations of Iron or Chromium, the respective d-band density functions will dominate, but as the elements mix, the s-band density will also contribute.

$$V(r) = \sum_i a_i (r - r_i)^3 H(r_i - r) \text{ where } H \text{ is the Heaviside Step function} \quad (5.69)$$

$$\phi_d^{CrCr}(r) = \sum_k b_i (r - r_k)^3 H(r_k - r) \text{ where } H \text{ is the Heaviside Step function} \quad (5.70)$$

$$\begin{aligned} \phi_s^{FeCr}(r) &= (H_s r^3 \exp(-\xi_s r))^2 \\ \phi_s^{FeFe}(r) &= 0.0 \\ \phi_s^{CrCr}(r) &= 0.0 \end{aligned} \quad (5.71)$$

The resulting potential, with the s-density fitted to the mixing enthalpy of Iron and Chromium, reproduces the formation of  $\alpha$ -prime phase Cr with thermal ageing over a range of Cr concentrations.

The Fe-Cr potentials were further developed by Bonny et al. The mixing enthalpy changes sign in Fe-Cr, having a positive mixing enthalpy for Cr concentrations above 10% and a negative mixing enthalpy below. This results in the solubility of Cr in the alloy at lower concentrations, but as the concentration rises there's a tendency for Cr to form  $\alpha$ -prime precipitates[114].

In the work by Bonny et al, the base Iron and Chromium potentials were those developed by Mendelev and Ackland, and from the Fe-Cr potentials of Olsson and Wallenius. In total, 11 functions make up the Bonny et al potential, higher than the usual 7 functions required for a two species EAM. In a standard EAM potential, the density contribution from an atom of one species is only dependent on that contributing atom, and not the embedded atom. This potential splits the density function into multiple bands and multiple permutations of atom species.

$$\begin{aligned}
 \rho_{AA}^d(r) &= \rho_{BA}^d = \rho_A^d \\
 \rho_{BB}^d(r) &= \rho_{AB}^d = \rho_B^d \\
 \rho_{AB}^s(r) &= \rho_{BA}^s \\
 \rho_{AA}^s(r) &= \rho_{BB}^s = 0
 \end{aligned} \tag{5.72}$$

The chosen s-band density function for the Bonny et al Fe-Cr alloy was selected as an exponential style function with a cut-off function.

$$\rho_{FeCr}^s(r) = kr^6 \exp(-2\xi r)g_{cut}(r) \tag{5.73}$$

$$g_{cut}(r) = \begin{cases} 1 & r \leq r_c^i \\ \frac{1}{2} \left( 1 - \sin\left(\frac{\pi}{2} \frac{r-r_m}{d}\right) \right) & r_c^i < r < r_c^f \\ 0 & r_c^f \leq r \end{cases} \tag{5.74}$$

$$F^s(\rho) = A_1\sqrt{\rho} + A_2\rho^2 \tag{5.75}$$

The magnetic properties of Chromium were considered during the development of these potentials. First-Principles calculations are equivalent to calculating properties at 0K, and at this temperature Chromium is antiferromagnetic. The Neel temperature for Cr, the point at which it transitions from an antiferromagnetic to a paramagnet, is 310K. The operating temperature of the alloys, within a reactor, will be above the Neel temperature. Chromium has a positive Cauchy pressure as a paramagnet, and a negative Cauchy pressure under the Neel temperature when an antiferromagnetic, and as a result of this and the operating temperature, a positive Cauchy pressure was used to fit the potential.

### Modified embedded-atom method

All the potential types considered so far are spherically symmetrical about the atom. Typically covalent bonds tend to be directional but metallic bonds, in contrast, are isotropic about each atom. However, in alloys and magnetic materials where the system is not isotropic, the modified embedded atom method may be used.

$$U_{MEAM,i} = \frac{1}{2} \sum_{j \neq i}^N V_{ij}(r_{ij}) + F_i(\vec{\rho}_i) \tag{5.76}$$

The original form for the embedding functional is given below (eq. 5.77)[115].

$$F(\vec{\rho}) = AE_{coh} \frac{\vec{\rho}}{\vec{\rho}^0} \ln \left( \frac{\vec{\rho}}{\vec{\rho}^0} \right) \tag{5.77}$$

The electron density is made from four contributing functions. The first,  $\rho^0$ , is not dependent on direction, but  $\rho^1$ ,  $\rho^2$  and  $\rho^3$  are dependent on the angle.

$$\begin{aligned}
\rho^0 &= \sum_i^N \rho_i^0(r^i) \\
(\rho^1)^2 &= \sum_{\alpha} \left( \sum_i \rho_i^1(r^i) \frac{r_{\alpha}^i}{r^i} \right)^2 \\
(\rho^2)^2 &= \sum_{\alpha} \left( \sum_i \rho_i^2(r^i) \frac{r_{\alpha}^i}{r^i} \frac{r_{\beta}^i}{r^i} \right)^2 - \frac{1}{3} \left( \sum_i \rho_i^2(r^i) \right)^2 \\
(\rho^3)^2 &= \sum_{\alpha} \left( \sum_i \rho_i^2(r^i) \frac{r_{\alpha}^i}{r^i} \frac{r_{\beta}^i}{r^i} \frac{r_{\gamma}^i}{r^i} \right)^2 - \frac{3}{5} \left( \sum_{\alpha} \frac{r_{\alpha}^i}{r^i} \rho_i^3(r^i) \right)
\end{aligned} \tag{5.78}$$

The partial electron densities may be combined as is eq. 5.79[115].

$$\begin{aligned}
\vec{\rho}_i &= \rho_i^{(0)} \cdot 2 / (1 + \exp(-\Gamma_i)) \\
\Gamma_i &= \sum_{h=1}^3 t^{(h)} \left[ \rho_i^{(h)} / \rho_i^{(0)} \right]^2 \\
\rho^{a(h)}(R) &= \exp(-\beta^{(h)}(R/r_e - 1))
\end{aligned} \tag{5.79}$$

This form of potential is implemented in LAMMPS and other molecular dynamics codes. It is more applicable to materials with bonds that depend on their angle, but it may have useful applications to metals where this is not as pronounced (if at all).

### 5.3.4 Function Types used by EAM and Two-Band EAM

In previous work, there are a wide range of functions used to represent the pair potential, density and embedding functional. These range from those that preserve and attempt to replicate the physics, to those that do away with knowledge of the physics underlying the potentials altogether. The pair potentials already covered here may also be used as the pair function component of the EAM or Two-Band EAM potentials.

A full list of the functions considered and included in the potential fitting code that resulted from this work is included in appendix ???. Only polynomial splines will be discussed further here.

#### Polynomial Splines

The form of polynomial spline used in the literature is a sum of N cubic polynomials that, by way of the Heaviside step function and their form, cutoff neatly at the desired radius. The cutoff radii are fixed and, during the fit of a potential, just the coefficients of each cubic spline are varied (eq. 5.84).

$$\begin{aligned}
V(r) &= \sum_i^N a_i (r - r_i)^3 H(r_i - r) \\
H(x) &= \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}
\end{aligned} \tag{5.80}$$

If a continuous second derivative is required the cubic spline may be replaced with a quintic spline (eq. 5.81).

$$V(r) = \sum_i^N a_i (r - r_i)^5 H(r_i - r) \quad (5.81)$$

### Polynomial Knot to Knot Splines

So far, analytic potentials have been discussed. There are existing potentials that do not have an analytic form, but are tabulated data that have the properties that they are continuous and have a continuous first derivatives. In attempting to fit or re-fit tabulated data, it would be problematic to adjust each data point individually given the number of data points and the requirement to have a continuous well behaved function with continuous first derivatives.

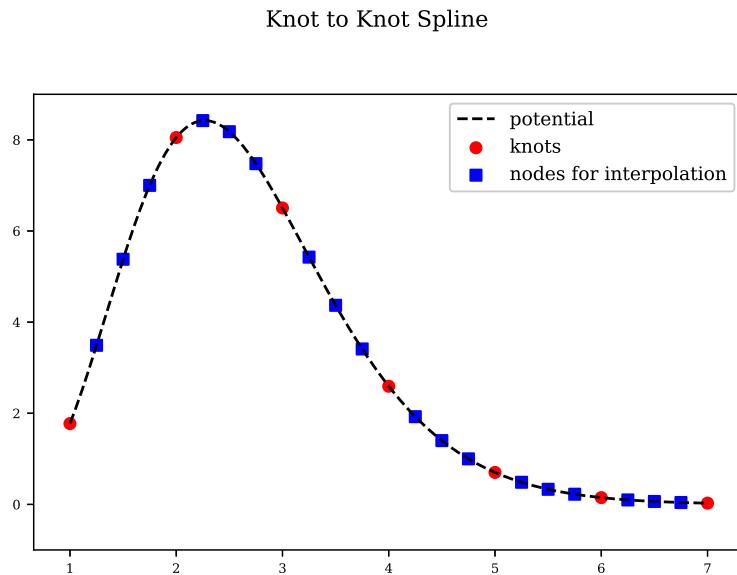


Figure 5.8: Polynomial knot to knot spline

The tabulated function is divided into sections, and the start and end of each section forms the knot of the spline. A polynomial may be splined between these knots, but the order of polynomial will depend on the data available. Immediately, the x and y positions are known:  $x_a, x_b, f(x_a), f(x_b)$ . A first order polynomial has two unknowns, and so these unknowns may be calculated using linear algebra.

$$\begin{aligned} c_0 + c_1 x_a &= y_a \\ c_0 + c_1 x_b &= y_b \end{aligned} \quad (5.82)$$

One of the requirements for the potential is to have a continuous first derivative, and this requires knowledge of the first derivative at each knot. In the figure, the knots are shown as circles. By taking several points near to each knot, shown as squares, the derivative at each knot may be calculated by interpolation. This allows a third order polynomial to be used (eq. 5.83).

$$\begin{aligned}
P_1 &= (x_A, f(x_A)) \\
P_2 &= (x_A, f'(x_A)) \\
P_3 &= (x_B, f(x_B)) \\
P_4 &= (x_B, f'(x_B))
\end{aligned} \tag{5.83}$$

$$\begin{aligned}
c_0 + c_1 x_a + c_2 x_a^2 + c_3 x_a^3 &= y_a \\
0 + c_1 + 2c_2 x_a + 3c_3 x_a^2 &= y'_a \\
c_0 + c_1 x_b + c_2 x_b^2 + c_3 x_b^3 &= y_b \\
0 + c_1 + 2c_2 x_b + 3c_3 x_b^2 &= y'_b
\end{aligned} \tag{5.84}$$

Rewriting as matrices:

$$\begin{bmatrix} 1 & x_a & x_a^2 & x_a^3 \\ 0 & 1 & 2x_a & 3x_a^2 \\ 1 & x_b & x_b^2 & x_b^3 \\ 0 & 1 & 2x_b & 3x_b^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} f(x_a) \\ f'(x_a) \\ f(x_b) \\ f'(x_b) \end{bmatrix} \tag{5.85}$$

It is important for the function and its derivative to be continuous. If it is also deemed necessary to have a continuous second derivative, a quintic spline may be used. By interpolating and calculating the second order derivative at each knot, a 5th order polynomial may be splined between pairs of knots.

This approach may be modified to spline functions of the form  $\exp(p(r))$  between knots, and this is useful for splining between a ZBL or similar repulsive function at small r and a function of a different form at larger r.

where

$$V(r) = \begin{cases} ZBL(r, q_1, q_2) & r \leq r_a \\ \exp(a + br + cr^2 + dr^3) & r_a < r < r_b \\ v(r) & r_b < r < r_{cut} \\ 0 & r \geq r_{cut} \end{cases} \tag{5.86}$$

To spline the exponential the equations are set up in a similar way, but this time the system of equations are non-linear.

$$\begin{aligned}
\exp(c_0 + c_1 x_a + c_2 x_a^2 + c_3 x_a^3) &= y_a \\
(c_1 + 2c_2 x_a + 3c_3 x_a^2) \exp(c_0 + c_1 x_a + c_2 x_a^2 + c_3 x_a^3) &= y'_a \\
\exp(c_0 + c_1 x_b + c_2 x_b^2 + c_3 x_b^3) &= y_b \\
(c_1 + 2c_2 x_b + 3c_3 x_b^2) \exp(c_0 + c_1 x_b + c_2 x_b^2 + c_3 x_b^3) &= y'_b
\end{aligned} \tag{5.87}$$

Newton-Gauss is used to solve this problem where initial parameters are set for the constants in the equation and varying until the function values and derivatives at point A and point B match the values being fitted to.

## Tending to Zero at the Cutoff Radius

It is desirable for the functions to be well behaved and continuous. The coulomb force between two charged particles reduces smoothly until theoretically at an infinite separation; it doesn't reach a set separation and abruptly drop to zero. It is impossible and unhelpful to consider a very large or infinite separation which is why the cutoff radius has been introduced. It represents the separation where the potential has reached zero; at the very least it's a trade off between accuracy and the computing time available for the problem.

The pair potential should therefore drop off to 0 at the cut off radius, and be equal to zero for larger separations. The electron density spherically around an atom should also smoothly drop off to zero, and similarly it should be equal to zero for distances larger than the cut off. The embedding energy is dependent on the density at the location of the atom embedded. The embedding function will not necessarily drop off to zero, and it depends on the density and not the separation.

In a molecular dynamics simulation, the neighbour list will usually be built considering atoms slightly more separated than the cut off radius of the functions. This allows the same neighbour list to be used for several time steps, before the need to update as atoms will get closer and further apart with each time step.

## Collisions and Interatomic Potentials

In a simulation at room temperature the average energy of the atoms in the simulation are approximately 0.05eV, and at temperatures near the melting point of iron the energy is approximately 0.2eV. These are energy ranges that interatomic potentials are designed to operate in, but in typical damage cascades in nuclear core components the atoms in the cascade, including the PKAs, have energies up to approximately 100KeV.

If a potential is not designed for use in collision simulations, it may not contain the data points required for such close separations. If a potential, such as the Buckingham potential (section 5.3.2), is used, the energy peaks at a certain separation and then drops once more. This would cause colliding atoms to stick to one another once they reached a certain separation.

The ZBL function is designed specifically with collisions in mind and should be used in MD simulations that involve collisions. In simulations of BCC tungsten[116], the material had an initial temperature of 10K and 523K with PKAs energies at 10keV, 20keV and 50keV. The models were simulated with MOLDY using an EAM potential where the pair potential is splined to a ZBL for short range interactions.

## 5.4 EAM Potential Fitting

### 5.4.1 Sacrificing Physical Elegance

We have no knowledge of a formula that can be applied to any material and describe the energy and forces between atoms exactly. Even first principles calculations rely on approximations, and the knowledge of an exact exchange functional eludes us. While it would be satisfying to derive a potential that reflects the fundamental physics, it may be something that is forever out of our reach.

It is more useful to develop a potential that replicates the properties of a material such as the relaxed lattice parameter, cohesive energy and elastic constants. The ZBL potential is an approximation to the repulsion between ions and is an appropriate potential to use to model high energy collisions, but not the bulk material. The Finnis-Sinclair are rooted in the second moment approximation of tight-binding theory and has been used to model metals.

Alternative potentials functions that lose physical significance may also be used to give potentials that work for specific elements under certain conditions. The trade-off for this loss any physical elegance [112] is a potential

that replicates the behaviour of a material under the conditions that are being studied, rather than use a more generic potential that isn't detailed enough to do so.

### 5.4.2 Exact Fitting to Parameters

As discussed in section 5.3.3 the functions may be fit directly to the known parameters for the metal. This was also the approach taken by Mishin whilst developing an alloy potential to model an Ni-Al system[117].

An exact fitting code was developed by G. Bonny in order to fit cubic spline type functions exactly to relaxed data including the Rose-Vinet equation of state (see section ??). Following work by Ackland[112], Mendelev[118], Olsson and Wallenius[113] the code developed by Bonny et al was used to develop a concentration dependent potential for Iron-Chromium[103]. More recently a potential has been fit for Tungsten-Rhenium alloys[119]. This code uses a quadratic program[120] to fit the potentials to the desired properties and it will be discussed in more detail in section .

### 5.4.3 Force Matching

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 [121] 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, typically calculated by first-principles calculations, and that given my the potential.

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

$$Z_F(\vec{\alpha}) = \sum_{k=1}^M \sum_{i=1}^k \sum_{j=1}^3 |F_{i,j}^k(\vec{\alpha}) - F_{i,j}^0|^2 \quad (5.88)$$

This may be extended to include the calculation of other properties:

- Lattice Parameter  $a_0$
- Cohesive Energy  $E_{coh}$
- Bulk Modulus  $B_0$
- Equation of State ( $E_0, V_0, B_0, B_0^p$ )
- Stress Tensor
- Elastic Constants
- Shear Modulus, Young's Modulus, Poisson Ratio
- Melting Temperature
- Surface Energy

Each may be weighted depending on how important the property is to the simulation the potential is required for.

$$Z(\vec{p}) = w_F Z_F(\vec{p}) + w_{b0} Z_{b0}(\vec{p}) + w_{e0} Z_{e0}(\vec{p}) + w_{a0} Z_{a0}(\vec{p}) + w_{ec} Z_{ec}(\vec{p}) + w_{ecoh} Z_{ecoh}(\vec{p}) \quad (5.89)$$

The result is a residual squared sum that measures how well or poorly a potential fits the required data.

#### 5.4.4 Pair and Embedding Symmetry Transforms

The combination of density function and embedding functional are not unique. When a potential is developed for a single element, this isn't a problem, but it must be considered when fitting to an alloy. If the density is based on real data, such as density distribution computed using a first principles code such as DIRAC, then there is some underlying meaning to the function. However, the functions fitted here have no real resemblance to the physics, other than having an effective cut off.

Bonny and Malerba[122] show two transforms that may be applied to the pair function, density function and embedding functional that the energy of any given atom is invariant to.

$$E_\alpha = \frac{1}{2} \sum_{\beta \neq \alpha} V(R_{\alpha\beta}) + F[\bar{\rho}_\alpha] \text{ where } \bar{\rho}_\alpha = \sum_{\beta \neq \alpha} \rho(R_{\alpha\beta}) \quad (5.90)$$

Given the usual definition of the EAM, in eq. 5.90, the first transform scales the value of the density function as simultaneously, and inversely, with the argument of the embedding functional (eq. 5.91). The second transform applies to the pair function and embedding functional (eq. 5.92).

$$\tilde{\rho}(R_{\alpha\beta}) = S\rho(R_{\alpha\beta})\tilde{F}[\bar{\rho}_\alpha] = F[\bar{\rho}_\alpha/S] \quad (5.91)$$

$$\tilde{F}[\bar{\rho}_\alpha] = F[\bar{\rho}_\alpha] + C\bar{\rho}_\alpha \tilde{V}(R_{\alpha\beta}) = V(R_{\alpha\beta}) - 2C\rho(R_{\alpha\beta}) \quad (5.92)$$

Mishin's approach, when fitting a binary alloy of Nickel and Aluminium, was to identify three transforms with three parameters that could be adjusted whilst fitting the alloy potential (eq. ??).

$$\begin{aligned} \tilde{\rho}_B &= s_B \rho_B(r) \quad \tilde{F}_B(\bar{\rho}) = F_B(\bar{\rho}/s_B) \\ \tilde{F}_A(\bar{\rho}) &= F_A(\bar{\rho}) + g_A \bar{\rho} \quad \tilde{V}_{AA}(r) = V_{AA}(r) - 2g_A \rho_A(r) \\ \tilde{F}_B(\bar{\rho}) &= F_B(\bar{\rho}) + g_B \bar{\rho} \quad \tilde{V}_{BB}(r) = V_{BB}(r) - 2g_B \rho_B(r) \end{aligned} \quad (5.93)$$

Bonny and Malerba chose to apply an effective gauge to the potentials for pure species. The gauge they chose sets the gradient of the embedding functional to zero at the equilibrium density, that of the relaxed crystal, as well as setting the equilibrium density to zero.

To transform the embedding functional such that  $F'[\bar{\rho}_{eq}] = 0$ , the constant  $C$  is set as  $C = -F'[\bar{\rho}_{eq}]$ . To transform the density such that  $\bar{\rho}_{eq} = 1$  the constant  $S$  is set as  $S = 1/\bar{\rho}_{eq}$ [122].

In the case of a binary alloy, the potentials for the pure elements are derived. Following this, the effective gauge transforms are applied to both pure element potentials. Finally, the cross pair potential is fit whilst the pure potentials are fixed.

## 5.5 Classical Molecular Dynamics

### 5.5.1 Introduction

Ab initio (DFT) calculations approximately solve the Schrödinger equation to calculate the energy, forces and stress of a given simulation. They take a relatively long time to calculate, have relatively small numbers of atoms (hundreds to thousands) and are fixed at one moment in time, although there are now programs that will run DFT MD. In comparison, Molecular Dynamics model a collection of atoms over a specified period of time. Much larger collections of atoms are possible (thousands to millions) and the interaction between atoms are predefined by interatomic potentials.

Molecular dynamics has been used to investigate the effects of radiation on materials. The damage event on the atomic scale is rapid in time, in the femtosecond to picosecond range, and affects a small volume of the material whereas the effects of the damage to the material on a mesoscopic and macroscopic scale may range from days to decades. In reactor pressure vessels, for instance, the damage and defect production will cover a large range of time and size scales[25]:

- $10^{-10}m$  to  $10^{-3}m$  ( $10^{18}$  to  $10^{25}$  atoms)
- $10^{-17}s$  to  $10^9s$  (tens of attoseconds to 30 years)

As a major element of steel, iron has often been the subject of MD simulations. Damage cascades have been studied in iron at a range of temperatures and PKA energies.

### 5.5.2 Molecular Dynamics Codes

Many MD codes are available for researchers to download freely and use from the Internet (although, terms may apply to the use). These include DLPOLY (Fortran), Large-scale Atomic/Molecular Massively Parallel Simulator (C++) and MOLDY (C). Most are able to run using message parsing interface to run the simulation in parallel across many compute nodes, with others being modified to take advantage of graphical processing units.

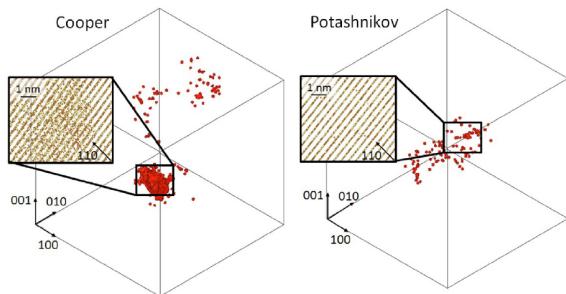


Figure 5.9: Damage in MOX (LAMMPS)[123]



Figure 5.10: Sputtering of plutonium oxide (LAMMPS)[124]

Damage cascades in mixed oxides have been modelled recently using the LAMMPS code. A  $70 \times 70 \times 70$  box with sides approximately 40nm in length and containing 4,116,000 atoms was used with PKAs having energies ranging from 5 to 75keV[123]. The sputtering of material from the surface of plutonium oxide (IV) has also been modelled with LAMMPS using a box of 393,216 atoms with a PKA of 87.7keV. The timescale of the sputtering simulation was 7.8 picoseconds ( $7.8 \times 10^{-12}$ ) and the entire event was split into  $1.0 \times 10^5$  time steps of  $7.8 \times 10^{-17}$ s.

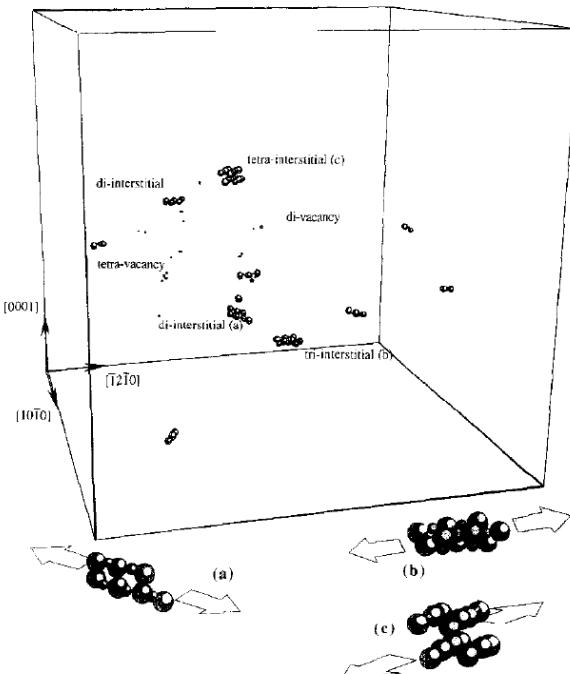


Figure 5.11: 5keV cascade (MOLDY)[125]

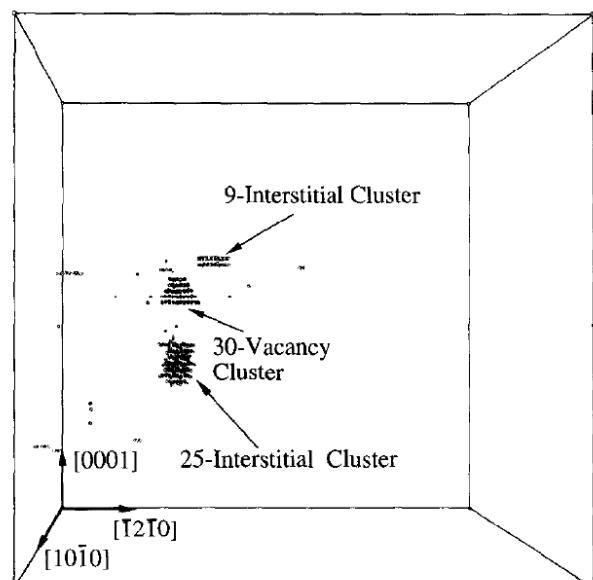


Figure 5.12: 20keV cascade (MOLDY)[125]

MOLDY has previously been used to model primary knock on atoms in metals, at energies of 5kev, 10keV and 20keV which would be slightly lower than the typical iron PKAs close to U235 fuel in the reactor (fig. 1.10). This simulation however was in alpha-zirconium and the higher mass of zirconium, relative to iron, would lead to lower energy PKAs. The damage cascades were modelled in a 104,832 atom block for the 5keV PKA (figure 5.11) and a 445,536 atom block for the 20keV PKA (figure 5.12).

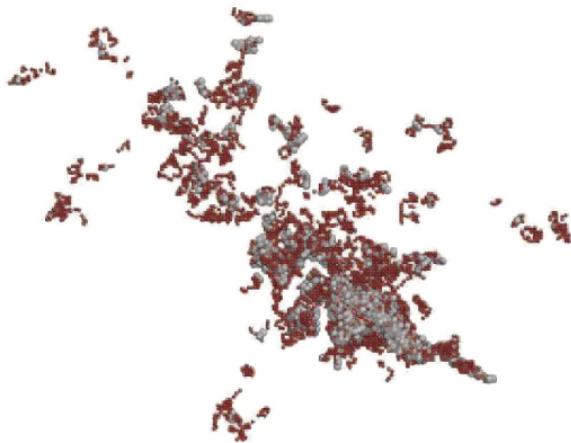


Figure 5.13: 100keV cascade after 50ps in TiO<sub>2</sub> (DL\_POLY)[126]

DL\_POLY has also been used to model damage cascades in various oxides (titanium, aluminium, silicon)[126]. The PKA energies are higher than those performed in MOLDY 15 years earlier, but still under 1MeV. The authors expect 1MeV simulation boxes to require at least 1 billion atoms, but the 100-200keV PKAs they modelled were within 10 million atom simulation boxes. Slightly larger boxes and damage cascades required longer simulation times, and with a box size of 43nm per side, a 50ps simulation time was used (fig. 5.13).

### 5.5.3 Calculating Energy, Force and Stress

Molecular dynamics simulations do not need to include the weak or strong force, and the gravitational force between atoms in the simulated material are so weak in comparison to the electromagnetic force, they can also be ignored. There is a force between all the atoms within a real material, but the electromagnetic force is inversely proportionally to the separation of the atoms. Above a certain separation, the electromagnetic force can also be ignored, in order to simplify the computation.

#### Neighbour List

A cut off radius can be introduced to limit the number of neighbours (section 5.3.4). As the lattice parameter decreases, the atoms are brought closer together, and as the cut off radius is increased more atoms are considered to be within the sphere of influence of one another. Both increase the number of neighbours each atom has (fig. 5.14).

Building a neighbour list may take a long time, depending on the parameters. If a simplistic approach is taken, for N atoms in the supercell, there will be  $27N^2$  checks between atoms to see whether they are within the cut off radius of one another. For larger numbers of atoms in a supercell, the whole may be decomposed into smaller domains.

For example, a 16x16x16 FCC supercell, containing 16,384 atoms, would require looping  $27 \times 16,384^2 = 7.25 \times 10^9$  times. Breaking the supercell into 64 smaller domains with 256 atoms in each, reducing the problem to  $64 \times 27 \times 256^2 = 1.13 \times 10^8$  loops.

For this work, the supercells used to calculate bulk properties from the interatomic potentials, and the configurations generated by DFT, contain fewer than 1,000 atoms, removing the need to decompose the configuration into smaller sub cells.

To build the neighbour list a halo configuration is created such that it lies on top of the real atoms, but extends at least as far as the cut off radius on each side of the real simulation box (fig. 5.15). Periodic boundary

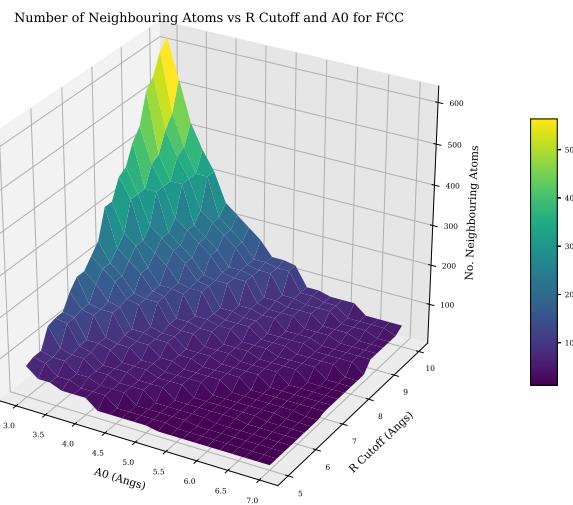


Figure 5.14: Neighbouring atom count dependant upon lattice parameter and cutoff radius

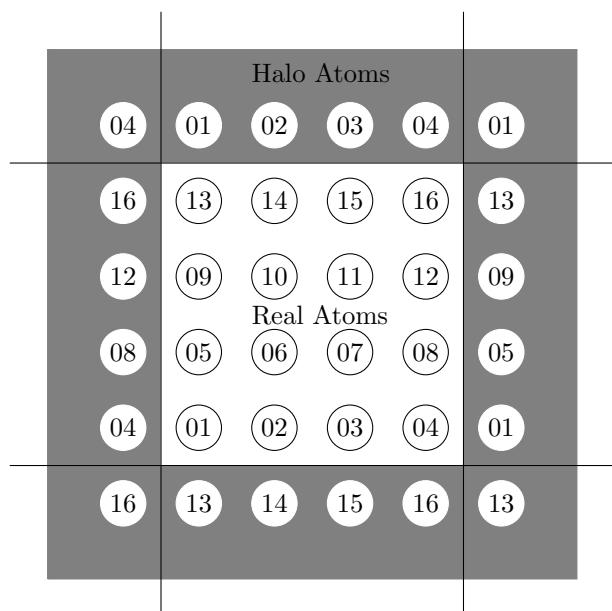


Figure 5.15: Atoms in simulation box with a halo of atoms at the boundary

conditions are used to construct the halo.

Once the halo list is computed, the neighbour list may also be computed by looping through the real atoms and halo atoms. A simple pseudo coded sub routine is given (listing 5.1).

Listing 5.1: Simple subroutine for generating a half size neighbour list

---

```

1 // real_atoms - array holding coordinates of all real atoms
2 // halo_atoms - array holding coordinates of all halo atoms
3 // real_ids - unique id for each atom
4 // halo_ids - unique atom ids for halo atoms
5 // nlist_ids - array to store ids
6 // nlist_r - array to store separation
7
8
9 // Define the cutoff and counter start value
10 r_cut = 5.0
11 nl_counter = 1
12
13 // Calculate square
14 r_cut_sq = r_cut ** 2
15
16 // Loop over all real atoms
17 DO n_real = 1, real_atom_count
18   // Loop over all halo atoms
19   DO n_halo = 1, halo_atom_count
20     IF (real_ids(n_real) .LT. halo_ids(n_halo)) THEN
21       r(1:3) = halo_atoms(n_halo, :) - real_atoms(n_real, :)
22       r_sq = SUM(r(1:3) * r(1:3))
23       IF (r_sq .LE. r_cut_sq) THEN
24         r_mag = sqrt(r_sq)
25         nlist_ids(nl_counter, 1) = real_ids(n_real)
26         nlist_ids(nl_counter, 2) = halo_ids(n_halo)
27         nlist_r(nl_counter, 1) = r_mag
28         nlist_r(nl_counter, 2:4) = r(1:3)/r_mag
29         nl_counter = nl_counter + 1
30       END IF
31     END IF
32   END DO
33 END DO

```

---

The resulting list will contain unique a pair combinations; i.e. the pair atom 1 and atom 2 will only be recorded once, and not also as atom 2 and atom 1.

Many modern MD codes are designed to run on computers with many thousands of processor cores. This leads to a choice of approaches, either to keep the configuration in shared memory and allow multiple processes to work on the same data, or break the data up, distributing it across various nodes.

In the case of DL\_POLY a link-cell domain decomposition is used[127]. The supercell is divided up into subcells, and these must be at least the size of the cut-off radius. Each processor will work on a geometric domain and these will only interact with their immediate neighbours. The halo data for each subcell is only passed between processors before and after the calculation of the equations of motion of the atoms within each subcell.

This is also the approach used by the LAMMPS code. The simulation is partitioned into either boxes where the density of atoms is uniform or tiles where there are regions with few or no atoms (fig. 5.16). This ensures processors are evenly utilised.

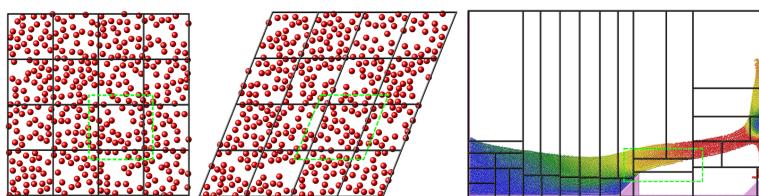


Figure 5.16: Partitioning in LAMMPS[128]

## Computing Total Energy

The total energy of the system is the sum of the individual energies for the atoms in the simulation. The type of atom (or pairs of atoms) will determine which function is used.

First, to compute the pair potentials:

- set the total energy of the system equal to zero
- set the starting energy for each atom in the system to zero
- loop through the atom pairs in the entire neighbour list
- for each atom pair, A and B, use the known separation and the potential function to compute the potential energy on atom A due to B and vice versa
- add this potential energy to both A and B
- after looping through the neighbour list, add all the energies due to the pair potential to the total energy

For an EAM or 2BEAM potential, the densities and embedding energies must also be computed:

- set the electron density at the position of (for each) atom to zero
- loop through the atom pairs in the entire neighbour list
- for each atom pair, using the density function for the atom A, the density at atom B due to atom A will be calculated and added to the density at atom B
- if the atoms are both of the same type the same density will be added to atom A due to atom B
- if the atom types are different, using the density function for the atom B, the density at atom A due to atom B will be calculated and added to the density at atom A
- following looping through the neighbour list, to calculate the densities at each atom, the list of atoms will be looped through
- for each atom, the density value at the position of that atom, will be input into the appropriate embedding function to calculate the embedding portion of the energy
- add all the embedding energies to the total energy of the system

For a 2BEAM potential, repeat the above procedure for the second group of density functions and embedding functions (do not repeat calculation of the pair potentials).

## Computing Forces on Atoms

In order to calculate the forces on the atoms with an EAM or 2BEAM potential, the neighbour list and atom list will need to be looped through several times. First, the pair potential and force due to the pair potential must be calculated by a complete loop through the neighbour list. At the same time, the density at each atom location is also calculated. The embedding energy may then be computed by looping through all the atoms and using the electron density at each atom to give the energy of the atom embedded in that density. The third and final loop will run through the neighbour pairs in the neighbour list once more computing the force on each atom due to it's embedding.

$$\begin{aligned} F_{pair}^k &= - \sum_{j=1, j \neq k}^N \frac{\partial V_{kj}(r_{kj})}{\partial r_{kj}} \frac{\vec{r}_{kj}}{|\vec{r}_{kj}|} \\ F_{embed}^k &= - \sum_{j=1, j \neq k}^N \left( \frac{\partial F}{\partial \rho_k} \frac{\partial \rho(r_j)}{\partial r_{kj}} + \frac{\partial F}{\partial \rho_j} \frac{\partial \rho(r_k)}{\partial r_{kj}} \right) \frac{\vec{r}_{kj}}{r_{kj}} \end{aligned} \quad (5.94)$$

The force on each atom, due to surrounding atoms, may be split into two; force due to the pair potential, and the force due to the embedding of the atom (eq. eq:computingForces)[111][127].

## Computing Stress

The stress is to be calculated assuming the system is at 0K[129] so the virial stress equation is eq. 5.95.

$$\tau_{ij} = \frac{1}{2V} \sum_{k,l \in V} \left( x_i^{(l)} - x_i^{(k)} \right) f_j^{(kl)} \quad (5.95)$$

Within the computer code, the individual force components between pairs of atoms are stored as well as an equal sized array of vectors representing their spatial separation. Those pairs with one atom in the halo around the simulation box are used to calculate the stress on the simulation box.

## Pseudo Code for the Energy, Force and Stress Subroutine

The calculation requires several arrays to temporarily store data. The electron densities and gradients of the density function are stored for multiple groups of densities and embedding functionals. The overall forces on each atoms also need to be stored as well as the force between pairs of atoms, as this is required to compute the virial stress. The Fortran code to compute this is included in the appendix (section ??) and a pseudocode in listing 5.2 explains the process.

Listing 5.2: Pseudo Code for Energy and Stress Force Calculation

---

```

1
2 electron_density[1:n_atoms, 1:bands] = 0.0 // electron density at each atom
3 energy = 0.0                                // total energy
4 forces[:, :] = 0.0                            // forces each atom, 3D
5 force_between_pairs[:, :] = 0.0               // used to store force between pairs, for stress computation
6 stress[:, :] = 0.0DO
7
8 // LOOP 1 - Pair energy, force and calculate densities
9 DO n = 1, neighbour_count
10    E = get_PairEnergy(atom_a, atom_b, r[n,:])
11    F[:] = get_PairForce(atom_a, atom_b, r[n,:])

```

```

12
13 // Save energy
14 energy = energy + E
15
16 // Save Force
17 f[atom_a, :] = f[atom_a, :] - F[:,]
18 f[atom_b, :] = f[atom_b, :] + F[:,]
19 force_between_pairs[:, :] = force_between_pairs[:, :] - F[:,] // Used to calculate stress
20
21 // Loop through density bands
22 DO band = 1, bands
23   // Electron density at A due to atom B
24   electron_density[atom_a, band] = get_Density(atom_b, r)
25
26   // Electron density at B due to atom A
27   electron_density[atom_b, band] = get_Density(atom_a, r)
28 END DO
29 END DO
30
31 // LOOP 2 - Embedding energy
32 DO n = 1, atom_count
33   // Loop through density bands
34   DO band = 1, bands
35     energy = energy + get_EMBEDDINGEnergy(n, band)
36   END DO
37 END DO
38
39 // LOOP 3 - Embedding force
40 DO n = 1, neighbour_count
41   // Loop through density bands
42   DO band = 1, bands
43     dgradA = get_DensityGradient(atom_b, r, band)
44     dgradB = get_DensityGradient(atom_a, r, band)
45     fgradA = get_EMBEDDINGGradient(atom_a, electron_density[atom_a, band])
46     fgradB = get_EMBEDDINGGradient(atom_b, electron_density[atom_b, band])
47
48     F[:,] = (fgradA * dgradB + fgradB * dgradA) * r[n, :]
49
50     f[atom_a, :] = f[atom_a, :] - F[:,]
51     f[atom_a, :] = f[atom_a, :] + F[:,]
52
53     force_between_pairs[n, :] = force_between_pairs[n, :] - F[:,] // Used to calculate stress
54 END DO
55 END DO
56
57 // LOOP 4 STRESS
58 DO n = 1, neighbour_count
59   // Only compute if the second atom is in the halo
60   IF(nlisthalo(n)) THEN
61     DO i = 1,3
62       DO j = 1,3
63         stress[i,j] = stress[i,j] + (r[n, i] * force_between_pairs[n,j])
64       END DO
65     END DO
66   END IF
67 END DO
68 stress[1:3, 1:3] = stress[1:3, 1:3] / (2.0 * volume)

```

### 5.5.4 Velocity Verlet Algorithm

Verlet integration is used by MD codes such as DL\_POLY, LAMMPS, Democritus as well as MD DFT codes, as in the case of CASTEP.

First, the forces at time step n are calculated  $\vec{f}_n$ . Now, using these forces and the velocity at time step n, the half step velocity is computed (eq. 5.96).

$$\vec{v}_{n+\frac{1}{2}} = \vec{v}_n + \frac{\vec{f}_n \delta t}{2m} \quad (5.96)$$

The position of the next step, n + 1, is calculated using the position at n, the time step  $\delta t$  and the half step velocity (eq. 5.97).

$$\vec{r}_{n+1} = \vec{r}_n + \vec{v}_{n+\frac{1}{2}} \times \delta t \quad (5.97)$$

The forces are recalculated at the new position  $\vec{r}_{n+1}$  to give  $\vec{f}_{n+1}$  and this, along with the half step velocity, is used to compute the velocity at step n+1,  $\vec{v}_{n+1}$  (eq. 5.98).

$$\vec{v}_{n+1} = \vec{v}_{n+\frac{1}{2}} + \frac{\vec{f}_{n+1} \delta t}{2m} \quad (5.98)$$

This process is repeated for each time step  $\delta t$  from the start of the simulation to the end.

### 5.5.5 Selecting a Time Scale

Take, for example, a 50x50x50 BCC Iron supercell with a lattice parameter of 2.87 angstrom. The supercell measures almost 15nm on each side. Simulating a thermal neutron, travelling at 2,200 m/s, would require an overall time period of at least 7 picosecond to capture the neutron passing through the supercell. Higher energy particles would require smaller overall time periods, divided up into as many time steps as the user requires.

If PKA damage is being modelled, there may be a point where the depth of the damage cascade becomes larger than the supercell itself. Damage cascades of 500KeV iron atoms into an iron target travel to a depth of up to 300nm, which would require a simulation supercell at least 1,000 BCC cells deep. Kinetic Monte Carlo would be better suited at simulating higher energy damage cascades.

To capture the required details the time steps would also need to be very fine, on the order of attoseconds or tens of attoseconds. This has been shown in the literature with time steps of  $7.8 \times 10^{-17}$ s[123].

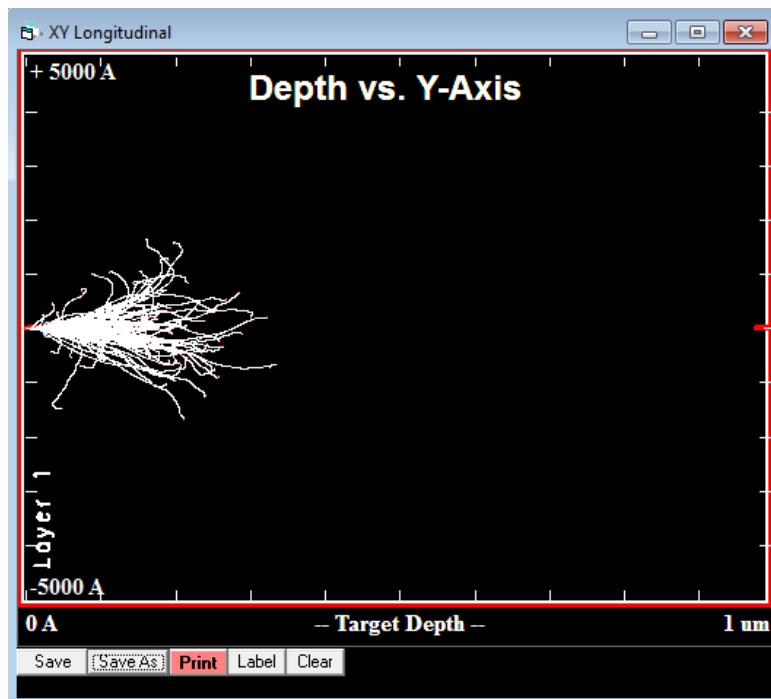


Figure 5.17: Damage cascade: 500kev iron projectiles into an iron target calculated with SRIM

## 5.6 Optimization

### 5.6.1 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 tasks where optimization plays a key role. 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-Marquardt Algorithm is used. Finally, a genetic algorithm and simulated annealing algorithm are coupled with LMA to locate the global optimal parameters for interatomic potential functions.

### 5.6.2 Continuous and Discrete Optimization

Problems that may be solved using discrete optimisation have, as the name suggests, discrete solutions. An example would be the travelling salesman problem, where the aim is to minimise the distance travelled by the salesman between cities. An example of the solution may be city E to A to B to C to D. There's a set of possible solutions to the problem. If the problem was changed such that any point within a 1 mile radius of the centre of each city is a valid solution, and the aim is to find the optimal points to travel between within the area of each city, it would become a continuous problem.

Fitting interatomic potentials to data is an example of continuous optimisation. The Morse potential takes three parameters plus the separation  $r$  between the two atoms. If the parameters are being optimised such that Morse potential matches the experimental forces (assuming they could be measured experimentally) between two atoms for a set of separation values, the choice of parameters isn't discrete; they are taken from a continuous range of real numbers.

### 5.6.3 Constrained and Unconstrained Optimization

There are constraints that may need to be considered, and constraints that apply to the fitting of interatomic potentials are as follows:

- continuous well behaved function
- continuous first-derivative, smooth change in force wrt  $r$
- positive electron density
- smooth cutoff at some  $r = r_{cut}$
- repulsion (ZBL/exponential) as  $r$  approaches 0
- continuous second-derivative
- enough parameters coupled with a good choice of functions so the potential is flexible enough to replicate all the experimental data well
- a small enough range of parameters to keep the parameter space small enough to search in a reasonable amount of time
- maintain physical elegance
- ignore physical elegance

Certain constraints listed will battle one-another, while others may be dropped, such as restricting to positive electron-densities if the choice to ignore physical elegance is selected.

### 5.6.4 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 near 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.

### 5.6.5 Global Optimization Algorithms

#### Simulated Annealing

A large amount of computing power is required today to solve many problems in Physics. The optimum arrangement of Iron atoms, at various concentrations and temperatures is one such example, but nature knows the correct structure it should take because it has access to the rule book. 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 algorithm was originally developed to improve upon the Metropolis algorithm, solving the issue of becoming stuck in non-global minima. Motivation for the algorithm included applying it to the optimum placement of chips on a circuit board[130], the travelling salesman (and similar) problem as well as finding the optimum arrangement of atoms.

Listing 5.3: Pseudocode for the simulated annealing algorithm

---

```

1 # Calculate the starting rss
2 rss = get_rss(p)
3
4 # Loop (n times, until a threshold is met, time runs out etc)
5 while(loop)
6 {
7     p_new = vary(p)           # Make a new set of parameters
8     rss_new = get_rss(p_new) # Get rss for these parameters
9
10    # Always accept BETTER
11    # Sometimes accept WORSE (depends on temperature and how bad it is)
12    if(rss_new < rss or rand(0.0,1.0) < exp((rss-rss_new) / temp))
13    {
14        p = p_new
15        rss = rss_new
16    }
17 }
```

---

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 (listing 5.3). The algorithm used in this work is available in the appendix (section ??).

## 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.

A set of solutions are bred with one another, and where there is an improvement the new variation is reinserted into the gene pool. There is a danger that the population can become a set of clones, so a large pool of sets of parameters is required, fresh parameters proposed every so often as well as a mechanism to prevent clones appearing.

As simulated annealing takes inspiration from cooling solids, genetic algorithms take inspiration from evolution. The algorithm used in this work is available in the appendix (section ??).

## Basin-Hopping

Lennard-Jones clusters are test systems of atoms models using the potential of the same name. Clusters of varying sizes have an optimal configuration with the minimum amount of energy. An algorithm was developed in the 1990s to improve upon existing global search algorithms and was applied to clusters of between 2 and 110 atoms[131]. The resulting basin-hopping algorithm allowed Wales and Doye to find three previously unknown minima.

$$\tilde{E}(\vec{X}) = \min(E(\vec{x})) \quad (5.99)$$

The objective of the algorithm was to solve eq. 5.99 where the vector  $\vec{X}$  is the coordinates of the atoms. A number of iterations, set by the user, are stepped through. The co-ordinates are randomly perturbed before the local minima is determined by using a local optimizer such as BFGS, LM-BFGS or CG[132]. Similar to SA, the minimized parameters are either accepted or discarded dependent upon how good the new parameters are.

### 5.6.6 Local Optimization Algorithms

#### Quadratic Programming (QP)

The method of Quadratic Programming, or quadratic optimization, is the task of minimising the matrix equation eq. 5.100 subject to imposed conditions[120].

$$\begin{aligned} \min_{\vec{x}} q(\vec{x}) &= \frac{1}{2} \vec{x}^T \vec{G} \vec{x} + \vec{x}^T \vec{c} \\ \text{subject to } a_i^T \vec{x} &= b_i, i \in \varepsilon \\ a_i^T \vec{x} &\geq b_i, i \in I \end{aligned} \quad (5.100)$$

This is the method used by Bonny et al. in their potential fitting program and will be discussed in section 5.7.

#### Gradient Descent

The gradient descent method is a reliable way of searching for a local minimum. It requires only the first derivative coupled with a line search to move closer to the minimum point. Close to the minimum, it is slower than the Newton-Gauss and LMA, but these require the computation of the Jacobian to estimate the Hessian matrix, so these are more computationally intensive. A pseudocode for the algorithm is available in the appendix (section ??).

#### 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 (eq. 5.101). 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.

$$\begin{aligned}\vec{J}^T \vec{J} \vec{p} &= \vec{J}^T \vec{r} \\ \vec{r} &= \vec{y} - f(\vec{x}, \vec{p})\end{aligned}\tag{5.101}$$

Close to a minima, the NG method quickly moves to the optimum point. Depending on the starting point, it may converge only to a local minimum, and not the global, and it may also be unstable, not converging at all.

### Levenberg Marquardt

The LMA is a NG 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 equation at the center of the algorithm is similar to that of NG (eq. 5.102).

$$\begin{aligned}\left( \vec{J}^T \vec{W} \vec{J} + \lambda \text{diag} \left( \vec{J}^T \vec{W} \vec{J} \right) \right) \vec{p} &= \left( \vec{J}^T \vec{W} \vec{r} \right) \\ \vec{r} &= \vec{y} - f(\vec{x}, \vec{p})\end{aligned}\tag{5.102}$$

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 preferential during their optimisation.

### Conjugate Gradient

In the method of steepest decent an initial value is selected and the gradient at that point is computed. This gives a search direction to look in and, following a line search, a new point is selected where the gradient is orthogonal to the search gradient. The process is iterative and at each new point the search gradient is replaced with the new orthogonal gradient (eq. 5.103[133]).

$$\begin{aligned}r_{(i)} &= b - Ax_{(i)} \\ \alpha &= \frac{r_i^T r_{(i)}}{r_{(i)}^T A r_{(i)}} \\ x_{(i+1)} &= x_{(i)} + \alpha_{(i)} r_{(i)}\end{aligned}\tag{5.103}$$

The conjugate gradient method is similar

$$\begin{aligned}
d_{(0)} &= r_{(0)} = b - Ax_{(0)} \\
\alpha_i &= \frac{r_{(i)}^T r_{(i)}}{d_{(i)}^T A d_{(i)}} \\
x_{(i+1)} &= x_{(i)} + \alpha_{(i)} d_{(i)} \\
r_{(i+1)} &= r_{(i)} - \alpha_{(i)} A d_{(i)} \\
\beta_{(i+1)} &= \frac{r_{(i+1)}^T r_{(i+1)}}{r_{(i)}^T r_{(i)}} \\
d_{(i+1)} &= r_{(i+1)} + \beta_{(i+1)} d_{(i)}
\end{aligned} \tag{5.104}$$

### Broyden Fletcher Goldfarb Shanno (BFGS)

## 5.7 Existing Potential Fitting Programs

### Bonny Quadratic Program Fitting

This code has been used to fit a number of potentials, including Fe-Cr and W-Re. It is a Fortran code and uses known properties of the material in different configurations. The only functional forms available in the 2012 version of the code are cubic spline for the pair and density functions and a polynomial for the embedding functional. The fitting subroutine uses quadratic programming to find parameters for the pair and embedding function splines.

### Hepburn FitPot

FitPot was created by Hepburn, a co-author of papers with Ackland[134]. It was written in Java and tackles the fitting as a non-linear least squares problem. The Levenberg-Marquardt Algorithm (LMA) is used to minimise the response function and uses the derivatives of the response function with respect to the input parameters to construct the Jacobian that is then used to construct an estimation of the Hessian.

### Brommer PotFit

Originally developed in C, the PotFit[135] program uses the force matching method of fitting. In the early versions of the code, splines were the only function choice available. In the years since, the option to fit analytic functions has been added. Simulated annealing is the primary search algorithm that attempts to locate the global minima. Powell's conjugate direction method is used to locate the local minima once the probable global minima is located. In order to use PotFit, a database of force, energy and stress for various configurations is required.

### Sheng Modified PotFit

A set of potentials have been derived by Sheng et al[136] and are available to download in tabulated form[33]. They were created using a modified version of the PotFit code by P. Brommer and F. Gähler[135]. The code had been modified extensively to include quintic spline interpolation, phonon calculation, elastic constant calculation and optimization techniques[33]. The potentials (typically) used 15 knot splines for pair and density functions and 6 knot splines for the embedding functional for each element, but the potentials were published as tabulated functions. Potentials for fourteen FCC metals were derived in the first instance, but more (including alloys) are

available on the website. Unfortunately, the (Sheng et al.) modified PotFit code is not available through the PotFit website[135], and the latest version of their code has moved on since 2011.

## 5.8 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.

### 5.8.1 Motivation For Using DFT In This Work

The use of interatomic potentials bridges the gap between the quantum scale and macroscopic scale. In order to develop these potentials either measurements or calculated values are needed for forces, energies and stresses. DFT is a convenient choice and has been used to fit potentials and calculate crystal properties many times (Sheng et al[136], Mehl et al.[100] Connetable and Thomes,[137] Ravindran et al[dfttisiravindran], Hepburn and Ackland[134] and more.

### 5.8.2 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.

### 5.8.3 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. As the DFT calculations in this work are static only the time-independent version will be discussed.

$$\hat{H}|\Psi\rangle = E|\Psi\rangle \quad (5.105)$$

The Hamiltonian  $\hat{H}$  is an operator and it is the total energy in the system.  $\Psi$  is the wave function (for example, the wave function of an electron) and this contains all the measurable information possible about whatever it represents.  $E$  is the energy eigenvalue of this system and this will depend on the eigenstate of the system. The wave function 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 (eq. 5.106).

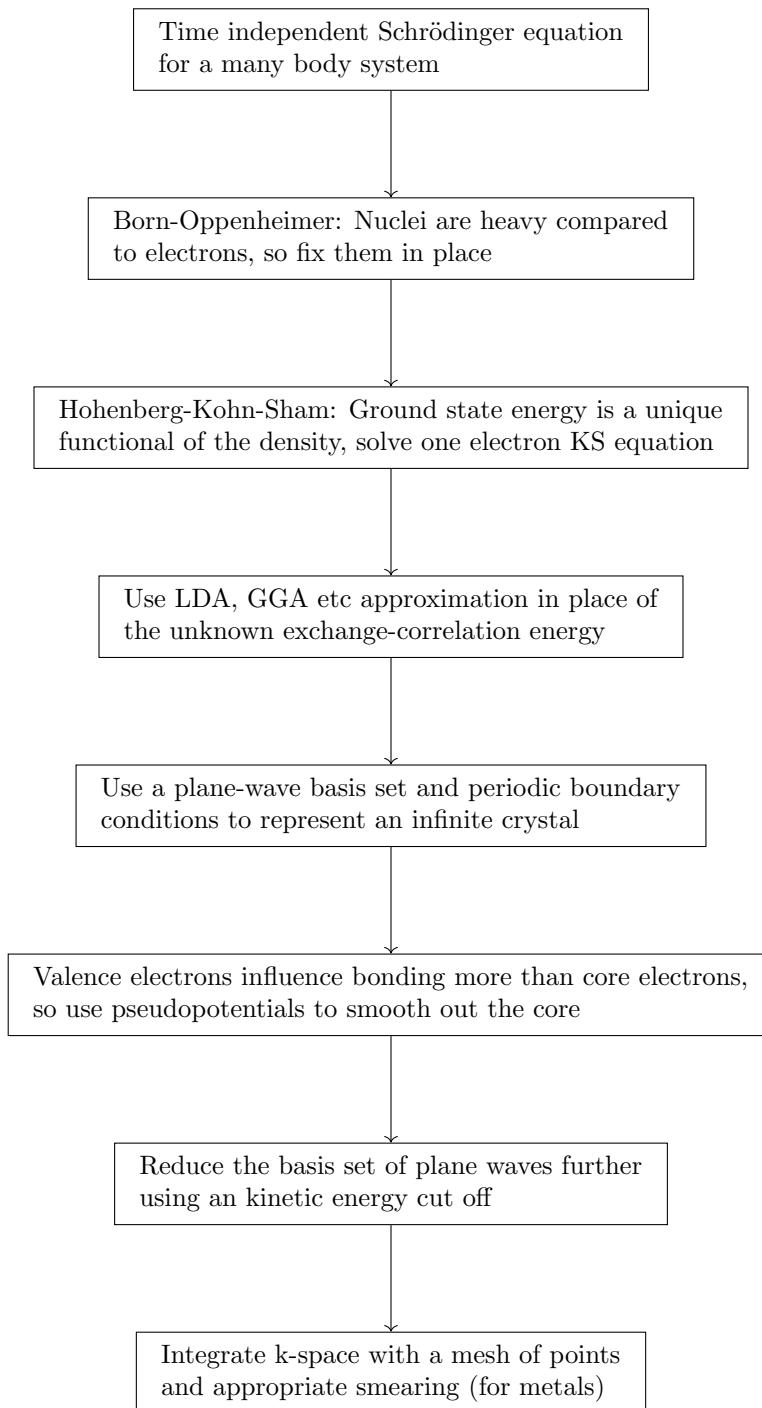


Figure 5.18: Common approximations used to enable DFT calculations

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Psi(x, y, z) dx dy dz = 1 \quad (5.106)$$

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.

$$\begin{aligned}\hat{H} &= \frac{\hbar^2}{2m} \nabla^2 \\ -\frac{\hbar^2}{2m} \nabla^2 \Psi(\vec{r}) &= E\Psi(\vec{r})\end{aligned}\tag{5.107}$$

If the particle is in a potential, it will have both kinetic energy  $\hat{T}$  and potential energy  $\hat{V}$  (eq. 5.108).

$$\begin{aligned}\hat{H} &= \hat{T} + \hat{V} = \frac{\hbar^2}{2m} \nabla^2 + V(\vec{r}) \\ \left[ -\frac{\hbar^2}{2m} \nabla^2 \right] \Psi(\vec{r}) &= E\Psi(\vec{r})\end{aligned}\tag{5.108}$$

### 5.8.4 Many Body TISE

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 towards this is to set up the Schrödinger 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  $1.0 \times 10^{-5}$  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  $1.0 \times 10^{36}$  times greater than the gravitational force, so it too can be neglected. Finally, the weak interacting force has a range of approximately  $1.0 \times 10^{-8}$  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 (eq. eq:TimeIndependentSchrodinger3).

$$\hat{H} = \hat{T}_e + \hat{T}_n + \hat{V}_{e-e} + \hat{V}_{e-n} + \hat{V}_{n-n}\tag{5.109}$$

The first two terms are the kinetic energy of the electrons and nuclei respectively (eq. eq:TimeIndependentSchrodinger4).

$$\begin{aligned}\hat{T}_e &= -\sum_i \frac{\hbar^2}{2m} \nabla^2 \text{Sum of kinetic energy of electrons} \\ \hat{T}_n &= -\sum_k \frac{\hbar^2}{2M} \nabla^2 \text{Sum of kinetic energy of nuclei}\end{aligned}\tag{5.110}$$

The last three terms are potential energy terms due to the electromagnetic force (eq. eq:TimeIndependentSchrodinger5).

$$\begin{aligned}V_{e-e} &= \frac{1}{2} \sum_{i,j, i \neq j} \frac{1}{|\vec{r}_i - \vec{r}_j|} \text{sum of potential energy between electrons} \\ V_{e-n} &= \sum_{i,k} \frac{z_i}{|\vec{r}_i - \vec{r}_l|} \text{sum of potential energy between electrons and nuclei} \\ V_{n-n} &= \frac{1}{2} \sum_{k,l, k \neq l} \frac{z_k z_l}{|\vec{r}_l - \vec{r}_k|} \text{sum of potential energy between nuclei}\end{aligned}\tag{5.111}$$

These operators are now input into the TISE (eq. 5.112).

$$\left[ \left( -\sum_i \frac{\hbar^2}{2m} - \sum_k \frac{\hbar^2}{2M} \right) \nabla^2 + \sum_{i,k} \frac{z_i}{|\vec{r}_i - \vec{r}_l|} + \frac{1}{2} \left( \sum_{i,j, i \neq j} \frac{1}{|\vec{r}_i - \vec{r}_j|} + \sum_{k,l, k \neq l} \frac{z_k z_l}{|\vec{r}_l - \vec{r}_k|} \right) \right] |\Psi\rangle = E|\Psi\rangle \quad (5.112)$$

### 5.8.5 Born-Oppenheimer

The TISE arrived at is far too complicated to solve, even for the simple ensembles of atoms. It represents the many body system of electrons and nuclei, and it takes the positions of all nuclei and electrons as input variables (eq. 5.113).

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

where  $\vec{r}_e = r_{e,1}, r_{e,2}, \dots, r_{e,i}$  (electron positions) (5.113)

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 for the electrons may be written with the electron co-ordinates as a variable, and the nuclei co-ordinates as a parameter (eq. 5.114).

$$\hat{H}_e(\vec{r}_e; \vec{r}_n) = \hat{T}_e(\vec{r}_e) + \hat{V}_{e-e}(\vec{r}_e) + \hat{V}_e - n(\vec{r}_e; \vec{r}_n) \quad (5.114)$$

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.

$$\hat{H}_e(\vec{r}_e; \vec{r}_n) \psi_e(\vec{r}_e; \vec{r}_n) = E_e(\vec{r}_n) \psi_e(\vec{r}_e; \vec{r}_n) \quad (5.115)$$

$$\Psi(\vec{r}_e, \vec{r}_n) = \chi_{ne}(\vec{r}_e) \psi_e(\vec{r}_e; \vec{r}_n) \quad (5.116)$$

The electronic Hamiltonian (eq. 5.115) is still dependent on the electronic co-ordinates. As there are three co-ordinates per electron, there are 3 dimensions when solving for a hydrogen atom. For an Iron atom, with 26 electrons, there are 78 dimensions, and for a 4x4x4 FCC supercell of Iron there would be 256 atoms, 26 electrons per atom and 3 dimensions per electron, giving a total of 19,968 dimensions. Even for small numbers of electrons, solving this equation is impractical.

### 5.8.6 Crystals, Reciprocal Space and Bloch Theorem

A Bravais lattice is a construct used to describe a periodic crystal lattice. It has the following properties given in eq. 5.117.

$$\vec{R} = n_1 \vec{a}_1 + n_2 \vec{a}_2 + n_3 \vec{a}_3$$

$$n_1, n_2, n_3 \in \mathbb{Z}$$

$$\vec{a}_1, \vec{a}_2, \vec{a}_3 \text{ are independent}$$
(5.117)

There are 14 Bravais lattices and 7 families of these lattices, and these lattices may be grouped by vector length and angle relationships (table. 5.4). This work is primarily concerned with cubic and tetragonal crystals, although distortions are applied to these crystals throughout.

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	$a, b, c$	$\alpha, \beta, \gamma,$

Table 5.4: Bravais lattice vector length and angle relationships

### Bloch Theorem

Metals are made up from grains which in turn are crystal lattices of atoms. Despite the majority of metals being composed of a large collection of microscopic crystals, rather than being a single perfect crystal, most of their properties may be calculated as if the metal were a single crystal.

The grain size of water quenched SS304 is approximately 30 micrometers across[138]. If the crystal were a cube, it would contain  $2.0 \times 10^{15}$  atoms and it would have sides over 100,000 atoms in length.

In solid state physics, in order to solve the TISE for such a crystal, it is useful to consider an infinite sized crystal. It may be helpful to visualise this as a "ring" of atoms in one dimension (fig. 5.19) or as a super-cell with periodic boundary conditions in three dimensions.

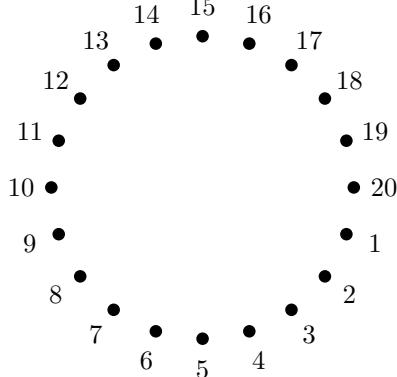


Figure 5.19: A useful, albeit incorrect, way of visualising an "infinite" chain of atoms in 1D

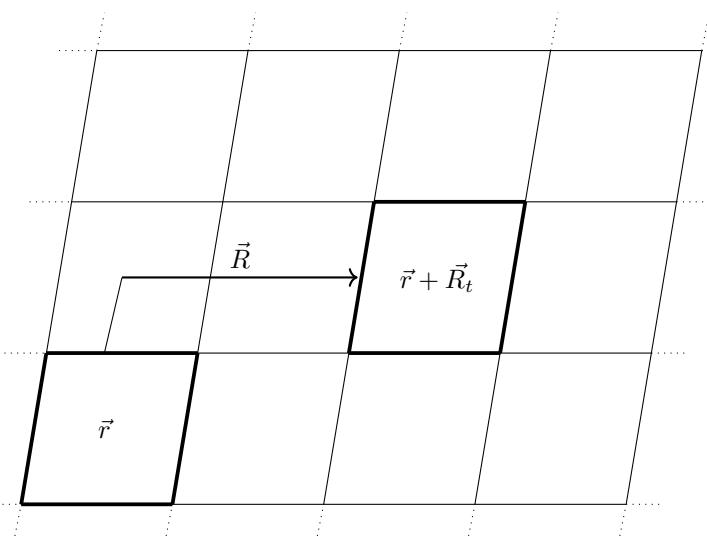


Figure 5.20: A 2D example of a translation by an integer multiple of the unit cell from the origin unit cell

As the structure is a repeating lattice, any point within a unit cell is equivalent to any other point translated by an integer multiple of the lattice vector (fig. 5.20).

$$\psi(\vec{r}) = \psi(\vec{r} + \vec{R}_t) \quad (5.118)$$

where  $\vec{R}_t = n_1 \vec{R}_1 + n_2 \vec{R}_2 + n_3 \vec{R}_3$

where  $\vec{R}$  is the real lattice vector

Reciprocal space, also known as k-space or momentum space, is a mathematical construct. It is an imaginary space where the lengths, and volumes, are the inverse of real space and planes of atoms are represented as points.

Starting with a lattice of points in real space  $\vec{R}$ , points in reciprocal space  $\vec{G}$  only are valid points if  $\exp(i\vec{G} \cdot \vec{R}) = 1$  [139]. A real space vector  $\vec{R}$  is transformed to its reciprocal in the following way, where  $\Omega$  is the volume of the primitive cell in real space.

$$\begin{aligned} \vec{g}_1 &= \frac{2\pi}{\Omega} \vec{r}_2 \times \vec{r}_3 \\ \vec{g}_2 &= \frac{2\pi}{\Omega} \vec{r}_3 \times \vec{r}_1 \\ \vec{g}_3 &= \frac{2\pi}{\Omega} \vec{r}_1 \times \vec{r}_2 \\ \Omega &= \vec{r}_1 \cdot (\vec{r}_2 \times \vec{r}_3) \end{aligned} \quad (5.119)$$

If the crystal structure is modelled as an infinitely repeating lattice, the potential also has the same periodicity (eq. 5.119).

$$V(\vec{r}) = V(\vec{r} + \vec{R}_t) \quad (5.120)$$

As the potential is a periodic function, it may be written as a Fourier series in reciprocal space (eq. 5.121).

$$V(\vec{r}) = \sum_{\vec{G}} V_{\vec{G}} \exp(i\vec{G} \cdot \vec{r}) \quad (5.121)$$

The wave function for an electron in the lattice is also periodic and may be written as the product of a plane wave.

$$\psi_{n\vec{k}}(\vec{r}) = u_{n\vec{k}}(\vec{r}) \exp(i\vec{k} \cdot \vec{r}) \quad (5.122)$$

$$\begin{aligned} u_{n\vec{k}}(\vec{r}) &= u_{n\vec{k}}(\vec{r} + \vec{R}_t) \\ \psi_{n\vec{k}}(\vec{r}) &= \psi_{n\vec{k}}(\vec{r}) \exp(i\vec{k} \cdot \vec{R}_t) \end{aligned} \quad (5.123)$$

The Born-von Karman boundary conditions apply to the wave function such that:

$$\psi_{n\vec{k}}(\vec{r} + \vec{R}_t) = \psi_{n\vec{k}}(\vec{r}) \text{ where } \vec{R}_t = \sum_i N_i \vec{a}_i \text{ and } N_i \in \mathbb{Z} \quad (5.124)$$

With these boundary conditions and use of plane waves, Bloch theorem replaces an enormous number of electrons with an infinite number electrons in a periodic lattice where only those in the unit cell need to be considered.

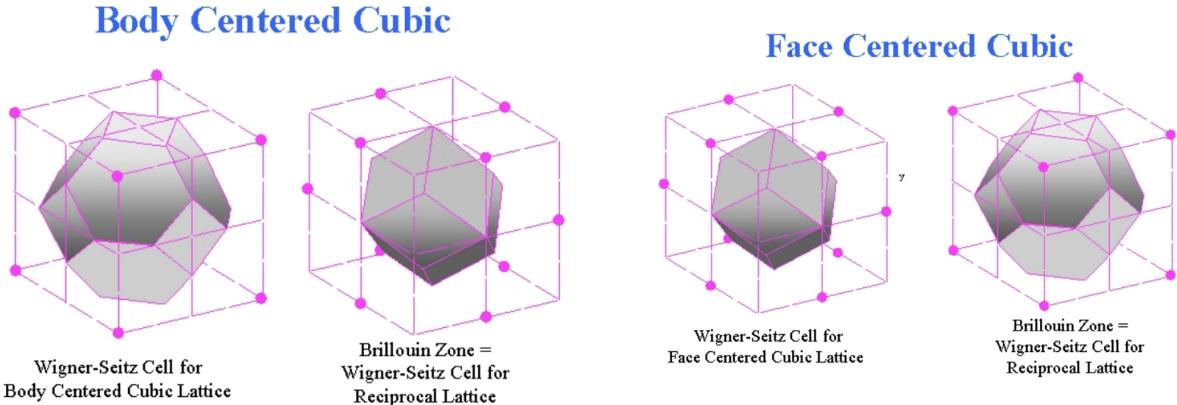


Figure 5.21: BCC Wigner Seitz Cell and BZ [140]

Figure 5.22: FCC Wigner Seitz Cell and BZ [140]

The (first) Brillouin Zone is a volume in (3D) reciprocal space and it is analogous to the Wigner-Seitz cell in real space. It is the primitive lattice cell in reciprocal space. The reciprocal of a BCC lattice is a FCC lattice (fig. 5.21) and vice versa (fig. 5.22). In real space a point anywhere in the lattice is a integer multiple of the lattice vector away from the same point in the origin Wigner-Seitz cell. There are an infinite number of wave vectors  $\vec{k}$  in reciprocal space, but by virtue of the periodicity of the lattice, they are also all found within the Brillouin Zone. If the crystal has symmetries, the BZ may be reduced further to the irreducible Brillouin zone.

Within the BZ there are energy surfaces. Picking a straight line through reciprocal space allows one to plot the energy bands along that one dimensional space. A particularly important energy surface is the Fermi surface and, in the one electron model, this marks the boundary between the occupied and unoccupied states at 0K [141]. The Fermi energy is the energy of the highest occupied state at 0K.

### 5.8.7 Hohenberg-Kohn Theorem

In the 1960s, Hohenberg and Kohn[142] simplified the problem of solving the many electron TISE in an exact way by proving:

- in an external potential  $v(\vec{r})$ , the potential is uniquely determined by the density of the ground state  $n_0(\vec{r})$  assuming that the particles are non-degenerate
- a uniquely defined functional  $E[\rho(\vec{r})]$  exists and the ground state energy,  $\min(E[\rho(\vec{r})])$ , is found by varying the density

The proof starts by picturing a box of electrons that interact with each other through coulomb repulsion within an external potential  $v(r)$ , for example the potential of “fixed” nuclei following the Born Oppenheimer approximation.

$$\begin{aligned}\hat{H} &= \hat{T} + \hat{V} = \frac{1}{2}\nabla^2 + V(\vec{r}) \\ \left[-\frac{1}{2}\nabla^2\right]\Psi(\vec{r})_0 &= E_0\Psi(\vec{r})\end{aligned}\tag{5.125}$$

$$\hat{H} = \hat{T} + \hat{V} + \hat{U}\tag{5.126}$$

where

$$T = \frac{1}{2} \int \nabla\psi^*(\vec{r})\nabla\psi(\vec{r})d\vec{r}\tag{5.127}$$

$$V = \int v(\vec{r})\psi^*(\vec{r})\psi(\vec{r})d\vec{r}\tag{5.128}$$

$$U = \frac{1}{|\vec{r}-\vec{r}'|}\psi^*(\vec{r})\psi^*(\vec{r}')\psi(\vec{r})\psi(\vec{r}')d\vec{r}d\vec{r}'\tag{5.129}$$

In the eqs. 5.126 5.127 5.126 5.129 the hamiltonian operator  $\hat{H}$  is a sum of the kinetic energy  $\hat{T}$ , the mutual coulomb repulsion  $\hat{U}$  and the operator due to an external potential (due to the nuclei)  $\hat{V}$ . The non degenerate ground state electron density is  $n(\vec{r}) = \langle\Phi|\phi^*(\vec{r})\phi(\vec{r})|\Phi\rangle$ .

The HK theorem shows that the external potential  $v(\vec{r})$  is a unique functional of the electron density  $n(\vec{r})$ . The proof is that by contradiction where by two different states are assumed to have the same ground state charge density.

#### State A

$$\begin{aligned}\Psi_A &\text{ with potential } V_A(\vec{r}) \\ \text{Hamiltonian } \hat{H} &= \hat{T} + \hat{V}_A + \hat{U} \\ \text{TISE } \hat{H}_A\Psi_A &= E_A\Psi_A\end{aligned}$$

Assumption - this state has the charge density  $n(\vec{r})$

#### State B

$$\begin{aligned}\Psi_B &\text{ with potential } V_B(\vec{r}) \\ \text{Hamiltonian } \hat{H} &= \hat{T} + \hat{V}_B + \hat{U} \\ \text{TISE } \hat{H}_B\Psi_B &= E_B\Psi_B\end{aligned}$$

The minimal property of the ground state gives the following[142][143]:

$$\begin{aligned} E_A &= \langle \Phi_A | \hat{H}_A | \Phi_A \rangle \\ E_A &= \langle \Phi_A | \hat{H}_B + \hat{V}_A - \hat{V}_B | \Phi_A \rangle \\ E_A &= \langle \Phi_A | \hat{H}_B | \Phi_A \rangle + \int d^3 \vec{r} n(\vec{r}) (v_A(\vec{r}) - v_B(\vec{r})) \end{aligned} \quad (5.130)$$

$$E_A > E_B + \int d^3 \vec{r} n(\vec{r}) (v_A(\vec{r}) - v_B(\vec{r})) \quad (5.131)$$

The indices A and B are interchanged to give a second equation:

$$\begin{aligned} E_B &> E_A + \int d^3 \vec{r} n(\vec{r}) (v_B(\vec{r}) - v_A(\vec{r})) \\ E_B &> E_A - \int d^3 \vec{r} n(\vec{r}) (v_A(\vec{r}) + v_B(\vec{r})) \end{aligned} \quad (5.132)$$

By adding eq. 5.131 and eq. 5.132 together the integrals will cancel out.

$$E_A + E_B > E_A + E_B \quad (5.133)$$

This (eq. 5.133) is a contradiction and proves the first part of the HK theorem:  $v(\vec{r})$  is uniquely determined by  $n(\vec{r})$ . If the charge density (fig. 5.23 and 5.23) is known, a single external potential exists for it.

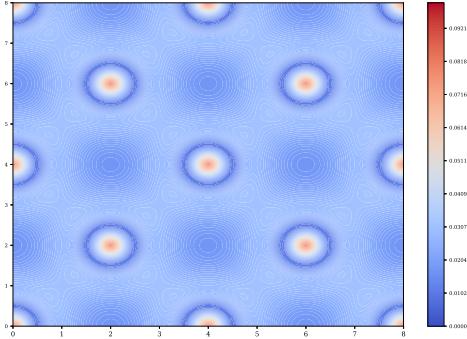


Figure 5.23: Charge density FCC aluminium xy plane at  $z = 0.0a_0$

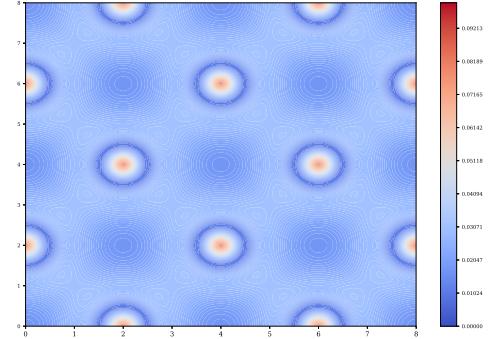


Figure 5.24: Charge density FCC aluminium xy plane at  $z = 0.5a_0$

The next part of the theorem is to show that the energy functional  $E[n(\vec{r})]$  exists and the minimum can be found by varying the charge density function. The kinetic and interaction functional  $\hat{T}$  and  $\hat{U}$  define a functional  $F[n(\vec{r})]$ .

$$F[n(\vec{r})] \equiv \langle \Phi | \hat{T} + \hat{U} | \Phi \rangle \quad (5.134)$$

This is a universal functional[142] and it makes up a part of the energy functional. The energy functional is comprised of the HK functional  $F[n(\vec{r})]$  and the external potential as shown in eq. 5.135.

$$E[n(\vec{r})] \equiv \int v(\vec{r})n(\vec{r})d\vec{r} + F[n] \quad (5.135)$$

For the ground state electron density  $n(\vec{r})$  the energy will be equal to the ground state energy  $E_0 = E_v[n]$ . Where N is the number of particles in the system, the integral of the density over all space will sum to N (eq. 5.136).

$$N[n] = \int n(\vec{r})d\vec{r} \quad (5.136)$$

If the system has N particles the energy level for a none ground state energy, k, is given in eq. 5.137.

$$\epsilon_{v,k}[\Psi_k] \equiv \langle \Phi_k | \hat{V} | \Phi_k \rangle - \langle \Phi_k | \hat{T} - \hat{U} | \Phi_k \rangle \quad (5.137)$$

This has a minimum where  $\Psi_k = \Psi_0$ , the ground state.

$$\begin{aligned} \epsilon_{v,k}[\Psi_k] &= \int v(\vec{r})n_k(\vec{r})d\vec{r} + F[n_k] \\ \epsilon_{v,0}[\Psi_k] &= \int v(\vec{r})n_0(\vec{r})d\vec{r} + F[n_0] \\ \epsilon_{v,k} &> \epsilon_{v,0} \end{aligned} \quad (5.138)$$

### 5.8.8 Kohn-Sham Equations

The year following the Hohenberg-Kohn theorem, a set of self-consistent equations were derived by Kohn and Sham. The ground state energy of interacting jellium in the potential of fixed nuclei, an external potential, can be written as follows[144]:

$$E = T_s + U + V_{n-e} + E_{xc} \quad (5.139)$$

$$E = T_s[\rho(\vec{r})] + \int d\vec{r}v(\vec{r})\rho(\vec{r}) + \frac{1}{2} \int \int d\vec{r}d\vec{r}' \frac{\rho(\vec{r})\rho(\vec{r}')}{|\vec{r} - \vec{r}'|} + E_{xc}[\rho(\vec{r})] \quad (5.140)$$

The kinetic energy,  $T_s$ , is that of a system of non interacting and  $E_{xc}$  is the exchange and correlation of an interacting system. The exchange and correlation energy functional exists, but is unknown.

The Kohn-Sham equations are used to calculate the energy of a many body Schrödinger equation by solving for a single electron.

$$\hat{H}_{KS}\psi_i = E_i\psi_i \quad (5.141)$$

$$\left(-\frac{1}{2}\nabla^2 + \hat{v}_{KS}\right)\psi_i = E_i\psi_i \quad (5.142)$$

$$\hat{v}_{KS}(\vec{r}_e, \vec{r}_n) = v_{n-e}(\vec{r}_e, \vec{r}_n) + \int d^3\vec{r}' \frac{\rho(\vec{r}'_e)}{|\vec{r}_e - \vec{r}'|} + v_{xc}[\rho](\vec{r}_e) \quad (5.143)$$

It is important to note that even though this is a one electron equation it is an exact solution.

### 5.8.9 Self-Consistent Solution

The Kohn-Sham equations cannot be solved in the usual way. They require a value for the density, but the density is obtained by solving the equations, so there is a dilemma. It can however be solved self consistently: an initial density is guessed, and this is repeatedly updated until the density in and out values are the same (or within a set convergence threshold). The basic algorithm used by PWscf from the Quantum Espresso suite is shown in fig. 5.25[145].

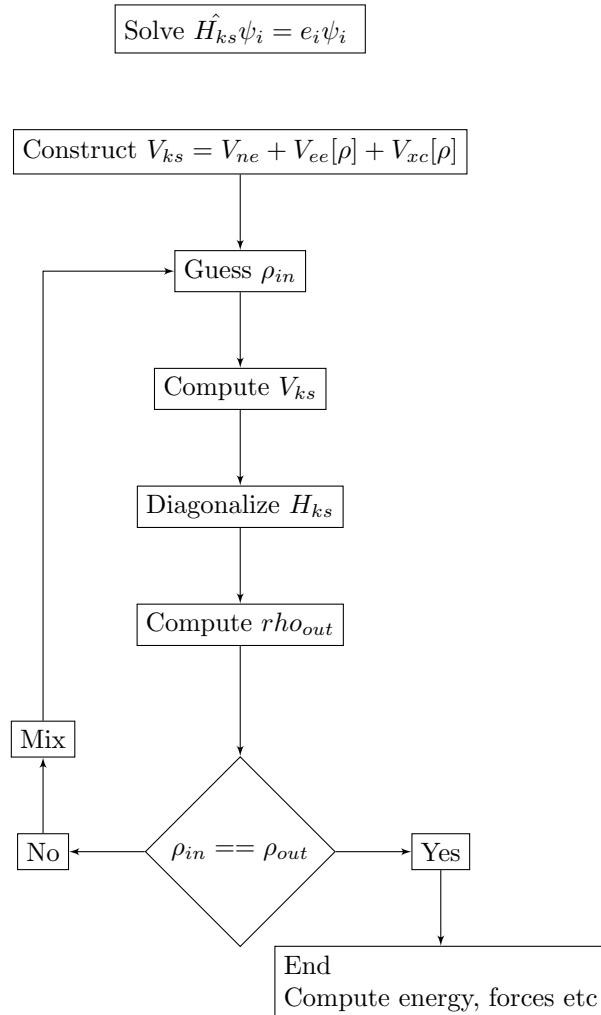


Figure 5.25: SCF algorithm for Quantum Espresso

In practice the input and output densities may never be exactly the same so rather than wait indefinitely for convergence, a threshold is set. The smaller the threshold the more iterations and time would be expected to pass, but the calculation will be closer to the solution.

With each iteration the density of the last iteration is mixed into a new density. If simple linear mixing is used a mixing beta determines the ration of new to old density as given in eq. 5.144[146].

$$\rho^{n+1} = (1 - \beta)\rho_{KS}^n + \beta\rho_{out}^n \quad (5.144)$$

More advanced mixing methods include Newton-Raphson or Broyden mixing. The PWscf program within Quantum Espresso uses Broyden for charge density mixing by default. Charge sloshing is an issue where convergence is slow or never happens. This is a particular problem for metals[147] and is more significant the larger the system[148]. The Thomas-Fermi screening used is based on the work by Johnson[149] and this works well for highly homogenous systems. Further work by Raczkowski, Canning and Wang[**localtfraczkowski**] developed a Pulay-Thomas-Fermi method that is implemented in PWscf with the local-TF mixing mode, and this is better suited to highly inhomogeneous systems.

### 5.8.10 Exchange-Correlation Energy

The Kohn-Sham equations have an exchange-correlation energy and this functional is used to collect together the electron energy not captured in the non-interacting energy functionals.

The Pauli exclusion principle states that two fermions, half integer spin particles, cannot occupy the same quantum state. This is why electrons occupy a unique orbital within an atom defined by the quantum numbers  $n, l, m_l$  and  $m_s$ . Consider two particles at points  $\vec{r}_a$  and  $\vec{r}_b$ ; the probability amplitude of the wavefunction of these particles equals 1.

$$|\Psi(\vec{r}_a, \vec{r}_b)|^2 = 1 \quad (5.145)$$

Exchanging the particles must also give the same result; they still must exist somewhere with probability 1.

$$|\Psi(\vec{r}_b, \vec{r}_a)|^2 = 1 \quad (5.146)$$

Bosons, integer spin particles, are symmetric when exchanged:

$$\Psi(\vec{r}_a, \vec{r}_b) = \Psi(\vec{r}_b, \vec{r}_a) \quad (5.147)$$

Fermions, on the other hand, are antisymmetric:

$$\Psi(\vec{r}_a, \vec{r}_b) = -\Psi(\vec{r}_b, \vec{r}_a) \quad (5.148)$$

By setting up a wavefunction for two Fermions, and exchanging them, it can be seen why they cannot exist in the same state.

$$\Psi_{ab} = \psi_1(\vec{r}_a, \vec{r}_b) - \psi_2(\vec{r}_b, \vec{r}_a) \quad (5.149)$$

$$\Psi_{ab} = \psi_1(\vec{r}_a, \vec{r}_b) - \psi_2(\vec{r}_a, \vec{r}_b) = 0 \quad (5.150)$$

The XC term combines the difference between the real system and the fictitious non-interacting system set out in the Kohn-Sham equations. It also includes the difference between the quantum mechanical electron-electron repulsion and classical electron-electron repulsion. Unfortunately, whilst we know a functional exists, we do not know the exact form of it[150].

### 5.8.11 Pseudopotentials

Plane wave basis sets are used to help solve the TISE. The DFT code used in this work is Quantum Espresso, and the binary that carries out the calculations is named PWscf: plane wave self consistent field.

Where  $\vec{G}$  is the reciprocal lattice vector, a summation of plane waves may be used to construct each electronic wave function (eq. 5.151)[151].

$$\Psi_{\vec{k},n} = \sum_{\vec{G}}^{|G| < G_{max}} c_{\vec{k} + \vec{G},n} \exp(i(\vec{k} + \vec{G})\vec{r}) \quad (5.151)$$

Unfortunately, the plane-wave basis sets required are too large to be used in practice, as the would need to represent the tight, rapidly oscillating inner orbitals as well as the valence electrons.

Bonding and material properties are primarily determined by valence electrons, not core electrons. 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.

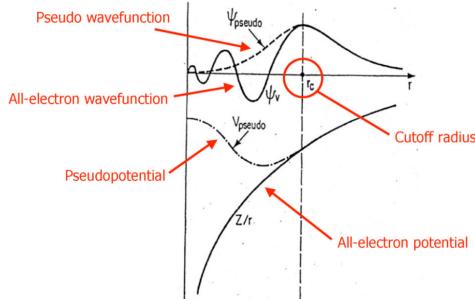


Figure 5.26: Replacing the complex potentials with a pseudopotential[152]

The eigen-energies of core electrons are also much larger than properties we see in materials, such as the cohesive energies and this also raises a concern that including the core electrons may introduce errors that are on a similar scale to the energies we are calculating.

The rapidly oscillating function in the core is replaced by a smoother pseudopotential. The same material properties are calculated as the valence electron functions are preferred, but a much small basis set is required to do so. There are a wide range of pseudopotentials available to use in DFT calculations, and one major distinguishing feature is how they treat the XC potential  $v_{xc}([n]; \vec{r})$ .

$$\left( \frac{1}{2} + V_{ext}(\vec{r}) + V_H(\vec{r}) + V_{xc}(\vec{r}) \right) \psi_{k,\sigma}(\vec{r}) = \epsilon_{k,\sigma} \psi_{k,\sigma}(\vec{r}) \quad (5.152)$$

The KS equation consists of several potentials including the exchange-correlation energy. This term represents the many-electron effects.

## LDA

The local density approximation replaces the electron density with jellium, a homogeneous electron gas. Kohn and Sham, in their original 1965 work, introduced jellium as the density model and, despite its simplicity in comparison to the real electron density distribution, it has worked well for simple metals.

$$E_{xc}^{LDA}[\rho] = \int d^3\vec{r}\rho(\vec{r})[e_x(\rho(\vec{r})) + e_c(\rho(\vec{r}))] \quad (5.153)$$

Several different LDA functionals have been developed over the years, including that of Perdew and Zunger. This functional takes the form given in eq. 5.154[153].

$$\epsilon_C[n(\vec{r})] = \begin{cases} \frac{-0.14231}{1+1.9529r_s(0.5)+0.334r_s} & r_s \geq 1 \\ -0.0480 + 0.0311lr(r_s) - 0.0116r_s + 0.0020r_sln(r_s) & r_s < 1 \end{cases} \quad (5.154)$$

## LSDA

Electrons are fermions and have half integer spin. When trapped in a potential, such as an atom or a crystal lattice, electrons must take difference quantum states and one of the parameters is the spin of the electron. The LDA does not treat spin at all, but the LSDA splits the density into up and down spin (eq. 5.155[150]).

$$\begin{aligned} \rho_{\uparrow}(\vec{r}) &= \sum_k^{occ} |\phi_{k,\uparrow}(\vec{r})|^2 \\ \rho_{\downarrow}(\vec{r}) &= \sum_k^{occ} |\phi_{k,\downarrow}(\vec{r})|^2 \\ \rho(\vec{r}) &= \rho_{\uparrow}(\vec{r}) + \rho_{\downarrow}(\vec{r}) \end{aligned} \quad (5.155)$$

The XC energy is now dependent on the spin up and spin down density as well as the position (eq. 5.156[150]).

$$\begin{aligned} E_{xc}^{LSDA}[\rho_{\uparrow}, \rho_{\downarrow}] &= \int d^3\vec{r}\rho(\vec{r})[e_x(\rho(\vec{r}))f(\zeta(\vec{r})) + e_c(\rho(\vec{r}), \zeta(\vec{r}))] \\ \zeta &= \frac{\rho_{\uparrow} - \rho_{\downarrow}}{\rho_{\uparrow} + \rho_{\downarrow}} \\ f(\zeta) &= \frac{1}{2} \left( (1 + \zeta)^{4/3} + (1 - \zeta)^{4/3} \right) \end{aligned} \quad (5.156)$$

The energy of the system now being solved is also dependent upon the charge density and spin polarization (eq. 5.157).

$$E = T[\rho, \zeta] + E_{ext}[\rho] + \frac{1}{2}E_{ee}[\rho] + E_{xc}[\rho, \zeta] \quad (5.157)$$

## GGA

In the LDA and LSDA approximations, the electron density is that of an homogeneous electron gas that does not represent the real electron density. The density of this gas may change in space, but locally it is a constant density with no gradient.

$$E_{xc}^{GGA}[\rho_\uparrow, \rho_\downarrow] = \int d^3r f(\rho_\uparrow, \rho_\downarrow, \nabla \rho_\uparrow, \nabla \rho_\downarrow) \quad (5.158)$$

The GGA functional (eq. 5.158 [150]) takes the spin up spin down electron densities as well as their gradients.

The PBE functional was developed to address some of the short falls of the Perdew-Wang PW91 LSDA functional. These include an over complication in the formulation of PW91 and that it was designed to satisfy many exact conditions. The PBE development focused more on energetically significant conditions[154].

$$E_c^{PBE}[\rho_\uparrow, \rho_\downarrow] = \int d^3r \rho [e_c(r_s, \zeta) + H(r_s, \zeta, t)] \quad (5.159)$$

$$E_x^{PBE}[\rho] = \int d^3r \rho e_x(\rho) F_x(s) \quad (5.160)$$

The functional is split into a correlation energy (eq. 5.159) and exchange energy (eq. 5.160).  $\zeta$  is the relative spin polarization  $\zeta = (\rho_\uparrow - \rho_\downarrow)/(\rho_\uparrow + \rho_\downarrow)$  and  $r_s$  is the local Seitz radius where  $\rho = 3/4\pi r_s^3$ . The functions  $e_x(\rho)$  and  $e_c(\rho)$  are the exchange and correlation energy per electron of unpolarised uniform electron gas[150]. A full description of the functional is in the appendix ??.

## LSDA vs GGA Potentials

Typically, GGA is an improvement over LSDA, with improvements in the total energies and structural energy differences[154]. Another short coming of LSDA is that it predicts FCC to be the optimal structure for pure iron at 0K, which is incorrect[155]. A list of success stories and failures were discussed shortly after the publication of the PBE functional and these are summarised in table ??[150].

In the case of the compound Titanium Disilicide,  $TiSi_2$ , GGA is in better agreement, overall, with experiment than LDA, although several values are better predicted by the latter. As shown in table 5.5, the data shows that LDA is within 1.26% of the experimental lattice parameters, 6.16% of the experimental bulk modulus and 18.3% the value of the experimental elastic constants; several of these were particularly poor, with an almost 45% disagreement. On the other hand, GGA is within 0.66% of the experimental lattice parameters, 3.62% of the experimental bulk modulus and 14.97% the value of the experimental elastic constants[105].

Lanthanum aluminate ( $LaAlO_3$ ) has also been studied using DFT, but this material is better modelled (to reproduce lattice parameters and bulk modulus) with either LDA or a specific form of GGA, the Perdew-Burke-Ernzerhof functional revised for solids (PBESOL) potential. The PBESOL is a revised PBE functional designed to better reproduce lattice parameters for solids, as shown in table 5.6.

A complete library of pseudo-potentials is available through the Quantum Espresso website. Several of these have been tested with the PWscf code to calculate the lattice parameter and bulk modulus of simple metals including Aluminium and Sodium.

## Successes of GGA

- atomization energy of molecules better
- binding energy curves more realistic
- Fe is bcc ferromagnet with GGA (fcc non-magnet with LSDA)
- Gives the anti invar effect in gamma-Fe and fc Fe-Mn
- Improvement on 4% accuracy for LSDA of alkali metal lattice constants
- Better lattice constants and bulk moduli of transition metals
- Isostructural transformation from open to close packed improved
- Successful calculation of a monovacancy in silicon
- Oxidation of the Si(001) surface using spin polarized GGA

## Failures

- exchange-correlation holes can be unrealistic under some circumstances
- works better for exchange correlation together than either alone
- interaction of electrons in different shells poorly described

Figure 5.27: Successes and failures of the GGA functional[150]

As can be seen from table 5.7, the PBE and PBESOL GGA type potentials are in general better at reproducing the lattice parameters. However, for Aluminium, the PZ LSDA type potential is in better agreement when calculating the bulk modulus. Overall, the GGA type potentials with the total self-consistent potential pseudized (these are the pseudopotential files that contain -n- in the name, rather than -nl-) perform best for these simple metals.

Whilst DFT is an exact method for solving the TISE, there are a number of approximations that still need to be made in addition to there being gaps in our knowledge (i.e. a lack of an exact functional for the XC).

### 5.8.12 Ecut, K-Point Integration and Smearing

As discussed in section 5.8.6 the wavefunction can be expressed as a periodic function with the same period as the crystal lattice (eq. 5.161). The periodic function may then be written as a sum of plane waves (eq. 5.162).

$$\psi_{n\vec{k}}(\vec{r}) = u_{n\vec{k}}(\vec{r}) \exp(i\vec{k} \cdot \vec{r}) \quad (5.161)$$

$$u_{n\vec{k}}(\vec{r}) = \sum_{\vec{G}} V_{\vec{G}} \exp(i\vec{G} \cdot \vec{r}) \quad (5.162)$$

Solving the Kohn-Sham equations requires the matrix diagonalization of a size NxN where N is the number of planewaves for each k-point in the Brillouin Zone[160]. The number of planewaves can be reduced to satisfy eq. 5.163. In practice this will mean reducing  $E_{cut}$  whilst ensuring the desired accuracy is still met.

$$\frac{\hbar^2 G^2}{2m} <= E_{cut} \quad (5.163)$$

Parameter	Experimental	LDA	GGA
a/angs	8.27	8.08	8.21
b/angs	4.80	4.74	4.81
c/angs	8.55	8.53	8.64
$B_0$ (Reuss)/GPa	146.8	156.9	141.9
$B_0$ (Voigt)/GPa	150.9	159.1	145.0
$C_{11}$ /GPa	317.5	377.2	326.0
$C_{22}$ /GPa	320.4	341.1	298.4
$C_{33}$ /GPa	413.2	425.3	371.9
$C_{44}$ /GPa	112.5	136.5	123.5
$C_{55}$ /GPa	75.8	93.7	85.3
$C_{66}$ /GPa	117.5	154.6	135.9
$C_{12}$ /GPa	29.3	27.8	22.4
$C_{13}$ /GPa	38.4	21.3	26.5
$C_{23}$ /GPa	86.0	95.1	105.5

Table 5.5: Experimental vs LSDA vs GGA for  $TiSi_2$ [105]

Parameter	Experimental	LDA	GGA	GGA (PBESOL)
a/angs	3.78	3.74	3.82	3.77
$B_0$ /GPa	215	224.00	190.35	206.41

Table 5.6: Experimental vs LSDA vs GGA vs GGA PBESOL for  $LaAlO_3$ [156]

In order to compute the energy using HK, the charge density in the BZ is calculated. It is impossible to diagonalise the matrix at an infinite number of k-points. The integral is replaced with a summation over selected points which in turn are weighted (eq. 5.164)[161].

$$\rho(\vec{r}) = \frac{1}{\Omega_{bz}} \sum_i \int f(\vec{k}, i) \psi_{i,\vec{k}}^*(\vec{r}) \psi_{i,\vec{k}}(\vec{r}) d\vec{k} \frac{1}{\Omega_{bz}} \int_{BZ} d\vec{k} \rightarrow \sum_{\vec{k}} \omega_{\vec{k}} \quad (5.164)$$

Due to Bloch theorem and the Born-Von-Karman boundary conditions, only the first BZ needs to be sampled. As discussed earlier, if there are further symmetries in the unit cell, only the IBZ need to be sampled.

The integral over k-space is replaced by a set of points in k-space that are sampled. The higher the number of points, the higher resolution the space is sampled in, but the longer the calculation will take. The Gamma point  $\Gamma$  is a high symmetry point in reciprocal space and is located at (0,0,0). It may be used as the only sampling point in some DFT calculations.

The Monkhorst-Pack grid is a special set of k-points. They are distributed evenly in reciprocal space and may be aligned such that either one point coincides with  $\Gamma$  or such that 8 points closest points surround  $\Gamma$  at an equal distance. By offsetting the grid from  $\Gamma$  and the coordinate axis of reciprocal space, the sampling better avoids points of high symmetry.

Where the system is spin degenerate, each state is occupied by 2 electrons for the lower  $\frac{N_e}{2}$  states ( $f(\vec{k}, i) = 2$ ), and zero otherwise ( $f(\vec{k}, i) = 0$ )[162]. This drop, from states occupied by two electrons to empty states occurs at the Fermi surface and it results in a discontinuity in the functions being integrated over in eq. 5.164. This is the case for an electron gas at absolute zero, but if the gas is heated then there is sufficient energy for electrons to occupy states above the Fermi energy.

Pseudo-potential	Element	a/angs	$B_0/\text{GPa}$
Experimental	Na	4.29[157]	6.3[157]
Na.pz-spn-kjpaw_psl.1.0.0	Na	4.06	8.72
Na.pbe-spn-kjpaw_psl.1.0.0	Na	4.20	7.67
Na.pbesol-spn-kjpaw_psl.1.0.0	Na	4.17	7.50
Experimental	Al	4.05[158]	76[158]
Al.pz-nl-kjpaw_psl.1.0.0	Al	3.98	78.6
Al.pz-nl-kjpaw_psl.1.0.0	Al	3.98	78.6
Al.pbe-nl-kjpaw_psl.1.0.0	Al	4.04	91.3
Al.pbe-nl-kjpaw_psl.1.0.0	Al	4.04	75.0
Al.pbesol-nl-kjpaw_psl.1.0.0	Al	4.01	87.7
Al.pbesol-nl-kjpaw_psl.1.0.0	Al	4.01	79.0

Table 5.7: Experimental vs LSDA vs GGA - the DFT values were computed with a PWscf[159] using a 2x2x2 cell, 7x7x7 kpoints and  $\text{ecutwfc}=50$

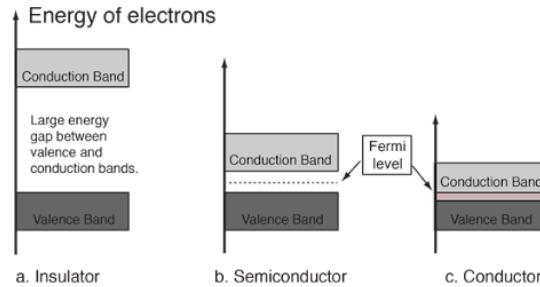


Figure 5.28: Energy gap between valence and conduction bands[163]

Fermi-Dirac statistics determine the probability of a Fermion having an energy  $E$  (eq. 5.165). As the temperature increases the probability of finding a fermion above the Fermi energy increases (fig. 5.29). This also smooths the discontinuity at the Fermi surface between the occupied and unoccupied states (at 0K).

$$F(E) = \frac{1}{\exp\left(\frac{E-E_F}{kT}\right) + 1} \quad (5.165)$$

To avoid having to integrate the BZ with a very fine mesh, a smearing function is introduced to remove the discontinuity at the Fermi surface, making the integral over the BZ differentiable at every point. The relationship between the delta and Heaviside step is used to replace the step function when integrated (eq. 5.166).

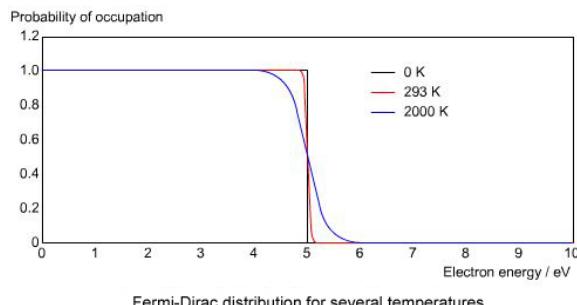


Figure 5.29: Probability  $F(E)$  of finding a fermion with energy  $E$  at several temperatures[164]

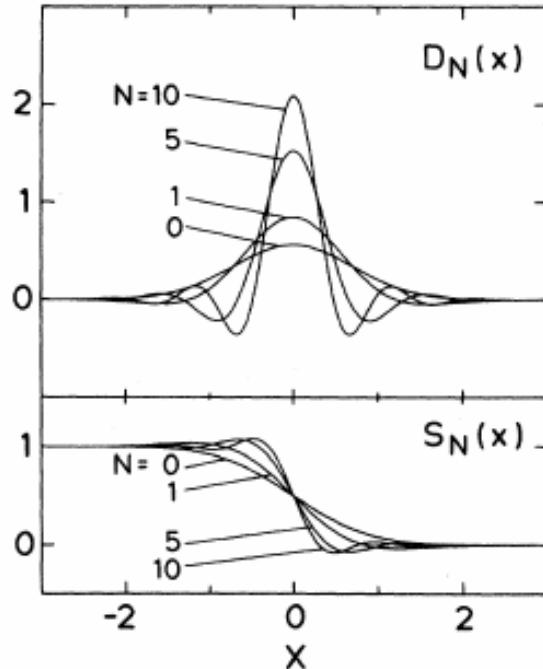


Figure 5.30: Delta function replaced by successive Hermite polynomials[165]

$$\int_{-\infty}^{\infty} \delta(k) dk = \Theta(x) \quad (5.166)$$

$$I = \int_{-\infty}^{\infty} S(\epsilon - E_F) \int_{BZ} f(\vec{k}) \delta(\epsilon - E(\vec{k})) d\vec{k} d\epsilon \quad (5.167)$$

The step function at the Fermi surface (eq. 5.167), that causes the discontinuity, may be replaced with a smearing function[165]. Four of these functions are available to use in the PWscf DFT code.

The Fermi-Dirac function, as mentioned above, is used to heat the system. It removes the discontinuity but the SCF will converge to the wrong energy (eq. 5.168).

$$S_{FD}(\epsilon) = \frac{1}{\exp\left(\frac{\epsilon-E_F}{kT}\right) + 1} \quad (5.168)$$

A Gaussian smear is also used to replace the step function although, unlike the Fermi-Dirac function, it has no physical meaning (eq. 5.169).

$$S_G(\epsilon) = \frac{1}{2} \left[ 1 - \text{erf} \left( \frac{\epsilon - \mu}{\sigma} \right) \right] \quad (5.169)$$

The Methfessel-Paxton method attempts to correct the errors introduced by the previous smearing functions where the delta function is replaced with Hermite polynomials (fig. 5.30 and eq. 5.171). A drawback of this method is that it allows for negative occupancy of electron states[162].

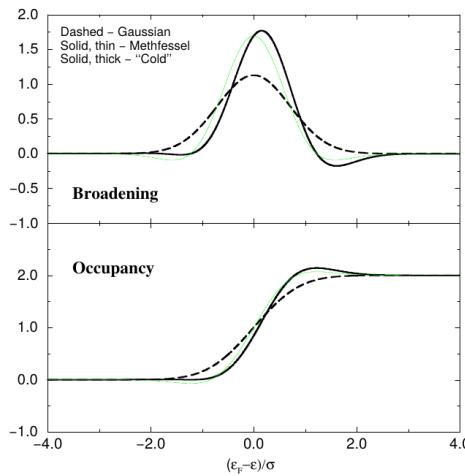


Figure 5.31: The occupancy for the Marzari-Vanderbilt smearing function is always positive, unlike the Methfessel-Paxton function[166]

$$x = \epsilon - E_F S_{MP,N}(x) = \frac{1}{2} [1 - \text{erf}(x)] + \sum_{n=1}^N A_n H_{2n+1}(x) \exp(-x^2) \quad A_n = \frac{(-1)^n}{n! 4^n \sqrt{\pi}} \quad (5.170)$$

The Mazari-Vanderbilt function was developed to address the short falling of the Methfessel-Paxton function. The occupancy where this function is used is always positive (fig. 5.31).

$$x = \frac{\mu - \epsilon}{\sigma} \delta(x) = \frac{1}{\sqrt{\pi}} \exp\left(-(x - (\frac{1}{\sqrt{2}}))^2\right) (2 - \sqrt{2}x) \quad (5.171)$$

### 5.8.13 Brief History of Magnetism

Magnetism as a phenomenon has been known to our civilisation for thousands of years. The magnetic mineral Lodestone contains the iron oxide Magnetite  $Fe_3O_4$  and it was used by the ancient Greeks over two thousand years ago.

The magnetic compass had been used since the 12th century, but it wasn't until the 17th Century that the Earth was discovered as a magnet itself by William Gilbert.

In the early 1800s, Hans Christian Oersted discovered that an electric current could move a compass needle, and this lead to much research into the connection between electricity and magnetism. Experimental work by Michael Faraday lead to the discovery that a changing magnetic field produces an electric field. By the mid-1800s the mathematical physicist, James Clerk Maxwell, derived his set of equations describing the connection between magnetism and electricity.

$$\partial \cdot \vec{E} = \frac{\rho}{\epsilon_0}$$

$$\nabla \cdot \vec{B} = 0$$

$$\nabla \times \vec{E} = -\frac{1}{c} \frac{\partial \vec{B}}{\partial t}$$

$$\nabla \times \vec{B} = \frac{1}{c} \left( 4\pi \vec{J} + \frac{\partial \vec{E}}{\partial t} \right)$$

### 5.8.14 Magnetism: Moving Charges and Spin

General Relativity and Quantum Mechanics were invented as theories to solve anomalies in science during the first part of the 20th Century, and they describe the very large and very small realms we do not notice in every day life. In describing how magnetism comes to be, it may be useful to use classical analogies, but that's all they are. Quantum Mechanics is required and this replaces intuition with a mathematical description.

$$\vec{B} = \frac{\mu_0 q}{4\pi r_{ij}^3} \vec{v} \times \vec{r}_{ij} \quad (5.172)$$

When a charged particle moves, it creates a magnetic field. The field at some point j created by a charge moving at  $\vec{v}$  at point i is calculated (eq. 5.172). Electrons do not orbit around a nucleus in the classical sense, but they do have an orbital angular momentum. The overall magnetic field created by an atom is made up of:

- orbital angular momentum of the electron
- spin of electrons
- spin of nucleons

The contribution of the nucleus to the magnetic field is small, and the orbital component and spin component of electrons are mostly responsible.

Spin is a property of quantum mechanics and it doesn't have an equivalent in classical mechanics, although it may be thought of similar to a spinning top or a rotating planet. Electrons are point like particles, and do not actually rotate, but they do have an intrinsic angular moment (spin). Where a spinning top might have a range of angular velocities, electrons have a quantised value and, as fermions, they have half integer spin.

### 5.8.15 Ferromagnetism, Antiferromagnetism and Hund's Rule

Electrons fill the shells of a ground state atom such that the energy is minimised. Four quantum numbers are used to describe electrons bound to an atom: n, l,  $m_l$  and  $m_s$ . The Pauli exclusion principle states that the electrons, which are fermions, cannot have the same quantum numbers as another electron bound to that atom.

Electrons may be spin up or spin down thus two electrons can have the same n, l,  $m_l$  with two values for  $m_s$  available. The energy is minimised by not pairing up and down electrons until all the free slots due to the coulomb interaction between the electrons. Once each slot contains one electron, they begin to pair.

The electronic configuration of Iron highlights this:

$1s^2$	$\underline{\uparrow}\underline{\downarrow}$		
$2s^2$	$\underline{\uparrow}\underline{\downarrow}$		
$2p^6$	$\underline{\uparrow}\underline{\downarrow} \quad \underline{\uparrow}\underline{\downarrow} \quad \underline{\uparrow}\underline{\downarrow}$		
$3s^2$	$\underline{\uparrow}\underline{\downarrow}$		
$3p^6$	$\underline{\uparrow}\underline{\downarrow} \quad \underline{\uparrow}\underline{\downarrow} \quad \underline{\uparrow}\underline{\downarrow}$		
$4s^2$	$\underline{\uparrow}\underline{\downarrow}$		
$3d^6$	$\underline{\uparrow}\underline{\downarrow} \quad \underline{\uparrow} \quad \underline{\uparrow} \quad \underline{\uparrow} \quad \underline{\uparrow} \quad \underline{\uparrow}$		

(5.173)

Rather than fill the 3d shell from left to right, the first five slots take electrons with spin in the same direction, and the remaining electron fills the first slot with spin opposite to the first five electrons.

Being fermions the wavefunction for the two electrons must be antisymmetric, with spins opposite to one another. Electrons with the same spin will repel one another and this causes an increase in the screening between the electrons and the nucleus. The screening lowers the attraction between the electrons and the nucleus that then results in the total energy of the atom decreasing[167].

Chromium

$$\begin{array}{c} 4s^1 \uparrow \\ 3d^5 \underline{\uparrow} \underline{\uparrow} \underline{\uparrow} \underline{\uparrow} \underline{\uparrow} \end{array} \quad (5.174)$$

Iron

$$\begin{array}{c} 4s^2 \uparrow\downarrow \\ 3d^6 \underline{\uparrow\downarrow} \underline{\uparrow} \underline{\uparrow} \underline{\uparrow} \underline{\uparrow} \end{array} \quad (5.175)$$

Cobalt

$$\begin{array}{c} 4s^2 \uparrow\downarrow \\ 3d^7 \underline{\uparrow\downarrow} \underline{\uparrow\downarrow} \underline{\uparrow} \underline{\uparrow} \underline{\uparrow} \end{array} \quad (5.176)$$

Nickel

$$\begin{array}{c} 4s^2 \uparrow\downarrow \\ 3d^8 \underline{\uparrow\downarrow} \underline{\uparrow\downarrow} \underline{\uparrow\downarrow} \underline{\uparrow} \end{array} \quad (5.177)$$

In pure Iron (eq. 5.175), under normal conditions, it is energetically favourable for the atoms to align in the same direction, and as such iron is ferromagnetic. Cobalt (eq. 5.176) has a face centered tetragonal structure at room temperature. It has a similar electronic structure to iron, but has three unpaired electrons in the 3d shell rather than four. It is a ferromagnetic whilst it remains in its CPH form. Nickel (eq. 5.177) is also ferromagnetic, but it is ferromagnetic in its FCC phase.

In its gamma phase, as it is in austenitic stainless steel, iron atoms align opposite to one another in an antiferromagnetic arrangement. This is also the same for BCC chromium (eq. 5.174) which is antiferromagnetic in its pure form under normal conditions.

### 5.8.16 Collinear and Non-Collinear DFT Calculations

# Chapter 6

## Methodology: Proton Activation and Radioactive Decay

*A computer program was developed to calculate the radioactivity of an ion irradiated target. The choice of projectile was limited to protons, but there is the option to increase this to other light ions. The decay equations were derived to be able to calculate the activity of isotopes within the target at time  $t$  during and after irradiation. The simulation is split into two parts: the first with the beam on, where there will be source terms for isotopes, and the second with the beam off where there are no external source terms.*

*SRIM, an ion transport code, was used to calculate ion trajectory data. Iron was irradiated experimentally by 36MeV protons with the University of Birmingham cyclotron, and its activity was measured and calculated using the activity code. Additional data where a colleague irradiated Molybdenum was also compared with that calculated by the activity code. Initially it was developed in Fortran, but the most recent version was written in Python.*

*The equations and code were then used to predict the activity of an iron sample irradiated to 100DPA for a range of proton energies. This is a reasonable damage dose that is expected in Gen IV reactors.*

### 6.1 Introduction

High flux neutron reactors are expensive to use, whereas proton accelerators capable of producing beams of adequate fluence and energy are more readily available, cheaper to buy and cheaper to run. Ion beams, due to the Coulomb interaction, are more controllable in terms of energy, direction and fluence. They may be concentrated on a desired target at a set fluence and energy.

Depending on the energy of the ion beam, the target material will become radioactive. The stable nuclei are transmuted and when the resulting isotope is unstable it will decay. An equation was derived to predict the activity of isotopes for any decay chain with source terms and branching factors included. Two versions of a computer code were developed to compute the reaction rates and activity of the irradiated targets.

## 6.2 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 6.1.

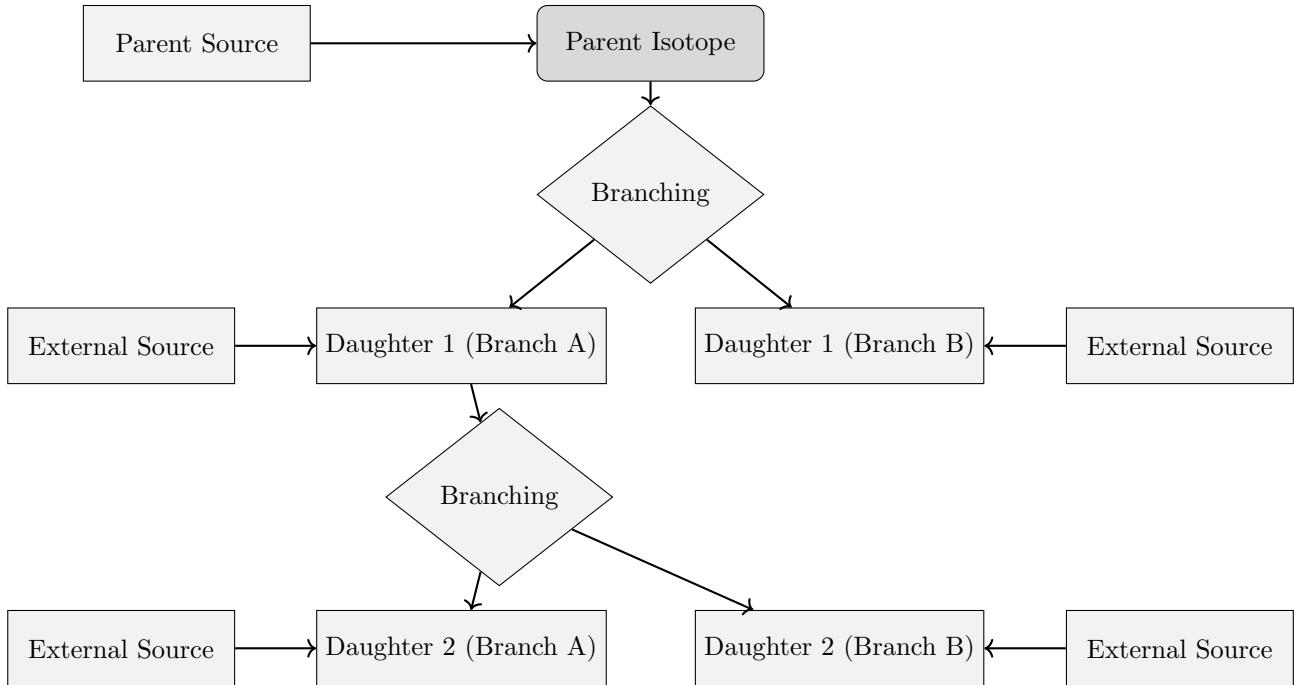


Figure 6.1: An example of several decay chains including branching factors and possible external source terms for each isotope on each chain.

### 6.2.1 Laplace Transform

Laplace Transforms (eq. 6.1) are a useful mathematical tool, and allow ordinary differential equations to be solved by simple algebraic manipulation in the s domain (appendix ??). Bateman took advantage of Laplace Transforms in deriving his equation, and this is the method that has been taken here as well.

$$F(s) = \int_0^{\infty} f(t) \exp(-st) dt \quad (6.1)$$

### 6.2.2 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  $\lambda_i$

- $\vec{b}$  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  $i^{\text{th}}$  isotope
- $N_i(t)$  amount of the  $i^{\text{th}}$  isotope at time  $t$
- $N'_i(t)$  change in amount of the  $i^{\text{th}}$  isotope, with respect to time, at time  $t$

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

$$N'_1(t) = \omega_1 - \lambda_1 N_1(t) \quad (6.2)$$

$$N'_i(t) = \omega_i + b_{i-1} \lambda_{i-1} N_{i-1}(t) - \lambda_i N_i(t) \quad (6.3)$$

$$N'_z(t) = \omega_z + b_{z-1} \lambda_{z-1} N_{z-1}(t) \quad (6.4)$$

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

$$N_1(s) = \frac{1}{s + \lambda_1} N_1(0) + \frac{1}{s(s + \lambda_1)} \omega_1 \quad (6.5)$$

$$N_i(s) = \frac{1}{s(s + \lambda_i)} (\omega_i) + \frac{1}{s + \lambda_i} (b_{i-1} \lambda_{i-1} N_{i-1}(s)) + \frac{1}{s + \lambda_i} N_i(0) \quad (6.6)$$

$$N_z(s) = \frac{1}{s^2} \omega_z + \frac{1}{s} b_{z-1} \lambda_{z-1} N_{z-1}(s) + \frac{1}{s} N_z(0) \quad (6.7)$$

### 6.2.3 Numerical Inversion of the Laplace Transform

The Gaver-Stehfest[168] 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 6.2).

$$f(t) \approx f_n(t) = \frac{\ln(2)}{t} \sum_{k=1}^{2n} a_k(n) F(s) \text{ where } n \geq 1, t > 0 \quad (6.8)$$

$$s = \frac{k \ln(2)}{t} \quad (6.9)$$

$$a_k(n) = \frac{(-1)^{(n+k)}}{n!} \sum_{j=\text{Floor}(\frac{k+1}{2})} j^{n+1} \binom{n}{j} \binom{2j}{j} \binom{j}{k-j} \quad (6.10)$$

The equation for the  $i^{\text{th}}$  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  $i^{\text{th}}$  isotope is reached (changing the equations appropriately for the parent, unstable daughter and stable daughter isotopes).

#### 6.2.4 Analytic Solution by Partial Fraction Expansion

The equation for the  $i^{\text{th}}$  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) (eq. 6.11).

$$\begin{aligned}
 N_4(s) = & \frac{1}{(s + \lambda_1)(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 N_1(0) \\
 & + \frac{1}{(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_3 b_4 \lambda_2 \lambda_3 N_2(0) \\
 & + \frac{1}{(s + \lambda_3)(s + \lambda_4)} b_4 \lambda_3 N_3(0) \\
 & + \frac{1}{(s + \lambda_4)} N_4(0) \\
 & + \frac{1}{s(s + \lambda_1)(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 \omega_1 \\
 & + \frac{1}{s(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_3 b_4 \lambda_2 \lambda_3 \omega_2 \\
 & + \frac{1}{s(s + \lambda_3)(s + \lambda_4)} b_4 \lambda_3 \omega_3 \\
 & + \frac{1}{s(s + \lambda_4)} \omega_4
 \end{aligned} \tag{6.11}$$

An example stable isotope, fourth (last) in the decay chain (including the parent isotope) (eq. 6.12).

$$\begin{aligned}
 N_4(s) = & \frac{1}{s(s + \lambda_1)(s + \lambda_2)(s + \lambda_3)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 N_1(0) \\
 & + \frac{1}{s(s + \lambda_2)(s + \lambda_3)} b_3 b_4 \lambda_2 \lambda_3 N_2(0) \\
 & + \frac{1}{s(s + \lambda_3)} b_4 \lambda_3 N_3(0) \\
 & + N_4(0) \\
 & + \frac{1}{s^2(s + \lambda_1)(s + \lambda_2)(s + \lambda_3)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 \omega_1 \\
 & + \frac{1}{s^2(s + \lambda_2)(s + \lambda_3)} b_3 b_4 \lambda_2 \lambda_3 \omega_2 \\
 & + \frac{1}{s^2(s + \lambda_3)} b_4 \lambda_3 \omega_3 \\
 & + \frac{1}{s^2} \omega_4
 \end{aligned} \tag{6.12}$$

By using partial fraction expansion and standard Laplace Transforms, the set of equations (eqs. 6.13, 6.14,

6.15, 6.16) is used to calculate the amount of the  $m^{\text{th}}$  isotope in the decay chain, providing the  $m^{\text{th}}$  isotope is unstable.

$$N_m(t; \vec{\lambda}, \vec{b}, \vec{w}, \vec{N}_0) = \sum_{k=1,m} r(k, m, \vec{\lambda}, \vec{b}) [f(t; k, m, \vec{\lambda}) N_{0,k} + g(t; k, m, \vec{\lambda}) w_k] \quad (6.13)$$

$$r(k, m, \vec{\lambda}, \vec{b}) = \begin{cases} \prod_{i=k, m-1} (b_{i+1} \lambda_i), & \text{if } k < m \\ 1, & \text{if } k = m \end{cases} \quad (6.14)$$

$$f(t; k, m, \vec{\lambda}) = (-1)^{m-k} \sum_{i=k, m} \left[ \frac{\exp(-\lambda_i t)}{\prod_{j=k, m; j \neq i} (\lambda_i - \lambda_j)} \right] \quad (6.15)$$

$$g(t; k, m, \vec{\lambda}) = \frac{1}{\prod_{i=k, m} \lambda_i} + (-1)^{m-k+1} \sum_{i=k, m} \left[ \frac{\exp(-\lambda_i t)}{\lambda_i \prod_{j=k, m; j \neq i} (\lambda_i - \lambda_j)} \right] \quad (6.16)$$

The set of equations (eqs. 6.17, 6.18, 6.19, 6.20) is used to calculate the amount of the  $m^{\text{th}}$  isotope in the decay chain, where the  $m^{\text{th}}$  isotope is stable.

$$N_m(t; \vec{\lambda}, \vec{b}, \vec{w}, \vec{N}_0) = N_m + w_m t + \sum_{k=1, m-1} r(k, m, \vec{\lambda}, \vec{b}) [f(t; k, m-1, \vec{\lambda}) N_{0,k} + g(t; k, m, \vec{\lambda}) w_k] \quad (6.17)$$

$$r(k, m, \vec{\lambda}, \vec{b}) = \begin{cases} \prod_{i=k, m-1} (b_{i+1} \lambda_i), & \text{if } k < m \\ 1, & \text{if } k = m \end{cases} \quad (6.18)$$

$$f(t; k, m, \vec{\lambda}) = \frac{1}{\prod_{i=k, m-1} \lambda_i} + (-1)^{m-k+1} \sum_{i=k, m-1} \left[ \frac{\exp(-\lambda_i t)}{\lambda_i \prod_{j=k, m-1; j \neq i} (\lambda_i - \lambda_j)} \right] \quad (6.19)$$

$$g(t; k, m, \vec{\lambda}) = \frac{1}{\prod_{i=k, m-1} \lambda_i} t - \frac{\sum_{i=k, m-1} [\prod_{j=k, m-1; j \neq i} \lambda_j]}{\prod_{i=k, m-1} \lambda_i^2} + (-1)^{m-k+1} \sum_{i=k, m-1} \left[ \frac{\exp(-\lambda_i t)}{\lambda_i^2 \prod_{j=k, m-1; j \neq i} (\lambda_i - \lambda_j)} \right] \quad (6.20)$$

### 6.2.5 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

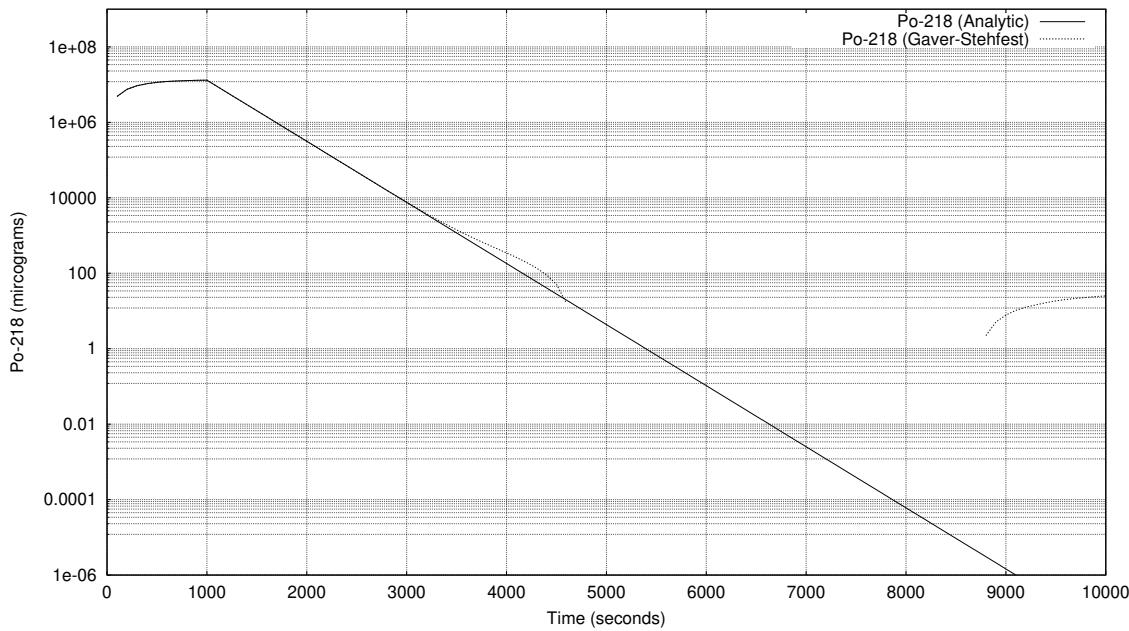


Figure 6.2: Decay of Po-218: Analytic and Gaver-Stehfest Calculations [22]

when the isotope decays away to a very small value. Figure 6.2 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 the numeric calculation drops below zero, whereas the analytic calculation remains above zero, as would be expected.

## 6.3 Computational Methods

### 6.3.1 Activity V1

The Activity V1 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. linear algebra).

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.

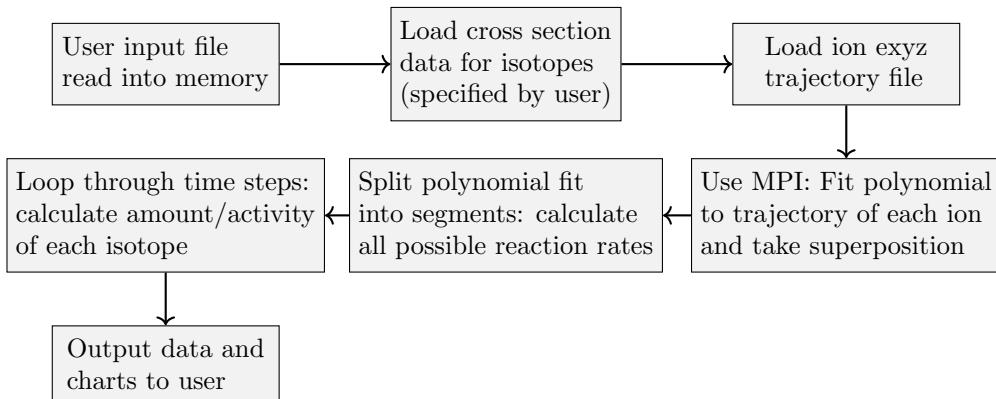


Figure 6.3: Flow chart of major processes in the Activity code

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 6.3 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.

A brief manual that accompanies the program is in the appendix section ?? and the published paper that accompanies the code is in section ???. The cross section data is taken from the existing TENDL-2013 datafile.

### 6.3.2 Activity V2

A second program was developed to use Python rather than Fortran. One problem that motivated this was occasionally a particular isotope would cause the code to crash, and a second issue was the complication of compiling the code for new users. The Python version runs with relatively few interventions by the user and it handles errors in the data more gracefully than the Fortran version. The procedure it follows is similar to the Activity V1 code and the manual is included in appendix section ??.

Due to a variation in range of energies in cross section data files, the cross section data was computed from scratch using the TALYS code and the BlueBEAR HPC.

### 6.3.3 Ion Trajectory Data

The SRIM code is used to generate ion trajectory data. This uses a quantum mechanical treatment of ion-atom collisions and statistical algorithms to speed up the process[169].

The trajectory data files used by both versions of the Activity code will be created using SRIM for the required beam energy and target combinations.

The target is set as pure Iron, and due to the way SRIM functions, the structure type is inconsequential. At just 0.5mm in thickness, the 36MeV ions pass through the targets and exit with an energy above 30MeV, and the exyz.txt data file covers this range of ion energies.

### 6.3.4 Activity Code Assumptions

A number of assumptions have been made for both codes to simplify the model. These may be addressed in future versions if development of the code continues. The flux of the beam is assumed to be constant at any depth into the material. Charged particle irradiation is ignored completely from the decay of radioactive isotopes in the target, and the amount of beta radiation may determine how the target is handled.

Gammas are assumed to be emitted isotropically and degrade only due to the increase in surface area as the distance from the source increases. No consideration is given to the dispersion of radioactive isotopes through the target material, how much material the gammas must pass through or how much air they must go through in order to reach the person (or measuring device).

These assumptions and possible improvements are considered in chapter ??.

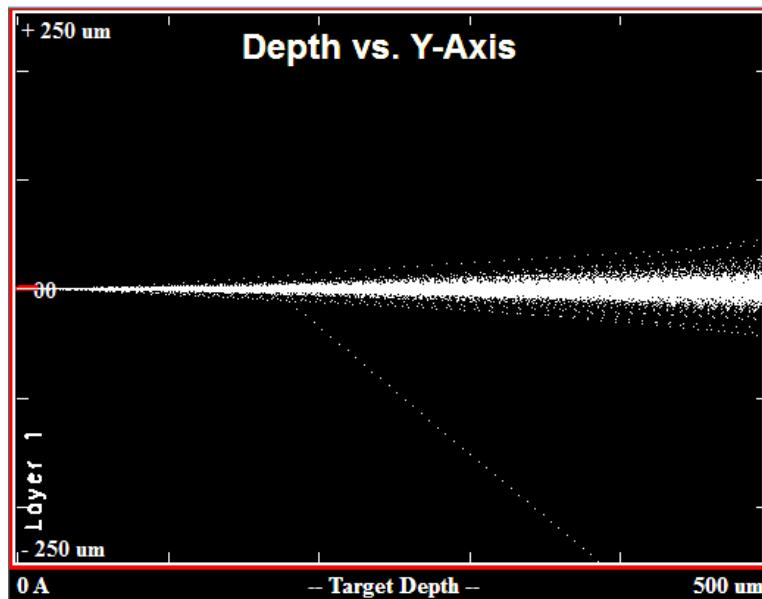


Figure 6.4: 36MeV Ion Track

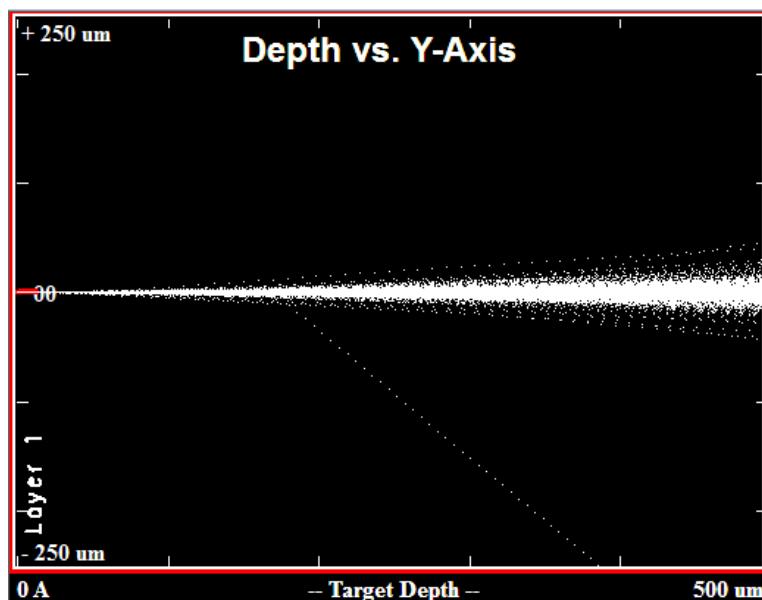


Figure 6.5: 36MeV Ion Track

### 6.3.5 Foil Activation by Neutron Irradiation

An additional feature was added to the Activity V2 code to estimate neutron activity 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 TALYS generated database 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.

## 6.4 Cyclotron Irradiated Iron

### 6.4.1 Cyclotron Beam Line

The Scanditronix MC40 cyclotron at the University of Birmingham has several beamlines and is capable of accelerating protons, deuterons, Helium 3 and Helium 4 with fluxes and energy ranges detailed in table 6.1. After running a simulation with SRIM, the 36MeV protons were expected to create an average of 15.8 vacancies each.

Particle	Energy (MeV)	Max Current (micro A)	Flux (ions per second)
p	8-40	60	$3.75 \times 10^{14}$
d	8-40	30	$1.87 \times 10^{14}$
$^4He^{2+}$	8-53	30	$9.36 \times 10^{13}$
$^3He^{2+}$	4-20	60	$1.87 \times 10^{14}$

Table 6.1: Beam Characteristics of the Scanditronix MC-40

The cyclotron is located in one room, and several beam lines run from the cyclotron to other rooms in the cyclotron building. The operation room is separate from the cyclotron and beam line rooms (figure 6.6).

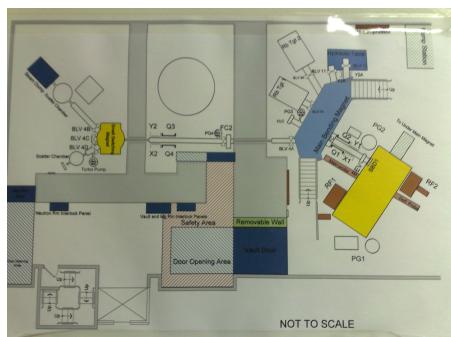


Figure 6.6: Cyclotron layout - main hall and beam lines

The iron sample was held perpendicular to the beam line, with the ions passing through the 0.5mm thickness of the sheet. The beam area was approximately  $6.4 \times 10^{-5} m^2$ , irradiating a volume of approximately  $3.2 \times 10^{-8} m^3$ . The target was irradiated for 300 seconds at 0.5 micro amps and it was expected to cause over  $1.4 \times 10^{16}$  displacements within the volume of iron targeted by the beam. With a number density of approximately  $8 \times 10^{28}$  atoms per cubic meter, giving a relatively low damage dose, when compared to that expected over the lifetime of a component within the reactor, of  $5^{-6}$  DPA.

During irradiation, the amount of gamma radiation released was enough to trip the alarm for the room (figure 6.7). The proton fluence was less than 1% of the maximum fluence capable of being produced by the cyclotron,



Figure 6.7: Gamma warning alarm during 0.5 microamp 36MeV proton irradiation of iron

so this was definitely a concern. To increase the damage dose, and keep to a shorter period of time, the current would most probably be increased to a much higher percentage, also increasing the rate of Gammas produced during irradiation.

#### 6.4.2 Measurement of Sample Activity

The sample was too radioactive to safely handle immediately after irradiation, so it was left to cool for several days before taking measurements. After it had cooled, a high purity germanium detector (figure 6.8) was used to measure the activity of the irradiated sample.



Figure 6.8: High Purity Germanium Detector

The detector and preamplifier are both cooled by liquid nitrogen to 77K, as the band-gap of germanium is 0.7eV and the thermal motion of electrons and nuclei at higher temperatures would induce currents causing noise and interference in the detector. As a gamma ray enters the detector it creates an electron-hole pair in the medium of the detector. The holes and electrons are attracted to the center and to the outer cylinder of the detector, or vice versa.

#### 6.4.3 Prediction of Activity

Following the measurement of the thin target, in the first instance the predicted activity will be estimated using the average ion energy, the cross section of this energy and the two reactions that result in Co-55 ( $^{54}_{26}Fe(p,\gamma)^{55}_{27}Co$ )

and  $^{56}_{26}Fe(p, 2n)^{55}_{27}Co$ . The activity code (version 1 and version 2) are then used to predict the activity of the sample as well as the expected gamma spectra.

# Chapter 7

## Results: Activity Code

The aim of the first part of this work was to derive an equation capable of calculating the radioactivity of an isotope and all the subsequent isotopes in the decay chain, taking into account branching factors and source terms.

This was achieved and two versions of a computer code, *Activity*, were developed. The first was written in Fortran and this was released in 2015. A revised version developed in Python3, with more features, has also been developed and is available for use.

The results of the computer code were compared to the experimental irradiation of Iron targets with the Scanditronix MC-40 Cyclotron. One code has also been used to predict the radioactivity of Iron targets against a range of ion energies up to 100DPA in order to select a beam energy that causes the required amount of damage, but minimises radioactive waste.

### 7.1 Extended Bateman Equations

The Bateman equations were derived from scratch to include a source term for each isotope and branching factors from parent isotope to daughter isotope(s). This process along with the numerical and analytic solutions are outlined in section 6.2.4.

The decay equations were programmed as a class in python (appendix ??) that used decay data accessible through a second class, isotopes, populated with data from the JEFF 3.1.1 data file. The static function for computing activity in this class was tested against numerically computed values.

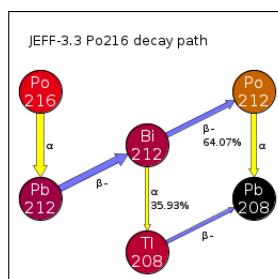


Figure 7.1: Decay path of Polonium-216

Due to the large range in magnitude of the half life for various isotopes, a decay chain was selected where the time step for the numeric calculation would be applicable to the half life of the isotopes in the decay chain,

whilst keeping the number of steps low enough to compute the activities in a reasonable amount of time with a simple algorithm. The decay path of  $^{216}_{84}Po$  (figure 7.1) contains 6 isotopes and includes one branch at  $^{212}_{83}Bi$ .

Isotope	Half Life (s)
Po-216	$1.45 \times 10^{-1}$
Pb-212	$3.83 \times 10^4$
Bi-212	$3.63 \times 10^3$
Po-212	$2.99 \times 10^{-7}$
Tl-212	$3.1 \times 10^1$
Pb-208	Stable

Table 7.1: Half life of isotopes in the Po-216 decay chain

The range of half life times (table 7.1) in the Polonium decay chain was from  $3.83 \times 10^4$ s for  $^{212}_{82}Pb$  down to  $2.99 \times 10^{-7}$ s for  $^{212}_{83}Po$  and the maximum number of steps used in the numeric calculation is  $10^{10}$ . This number of steps was near the limit at which increasing by another order of magnitude would take a significant amount of time to run the calculation.

Isotope	Source Rate	Starting amount $N_0$
Po-216	$2.0 \times 10^{-1}$	$1.0 \times 10^2$
Pb-212	$0.0 \times 10^0$	$5.0 \times 10^0$
Bi-212	$7.0 \times 10^{-2}$	$1.5 \times 10^1$
Po-212	$5.0 \times 10^{-3}$	$0.0 \times 10^0$
Tl-212	$0.0 \times 10^0$	$0.0 \times 10^0$
Pb-208	$1.0 \times 10^{-2}$	$3.0 \times 10^2$

Table 7.2: Parameters used for the decay equation vs numeric calculation comparison

The numeric code used to perform these calculations is detailed in the appendix (section ??). The calculation was set up with a range of source rates and isotope starting amounts including no source for  $^{212}_{81}Tl$  and a zero starting amount for both  $^{212}_{81}Tl$  and  $^{212}_{83}Po$ . The full details are given in table 7.2 and these were used in both the numeric and analytic codes.

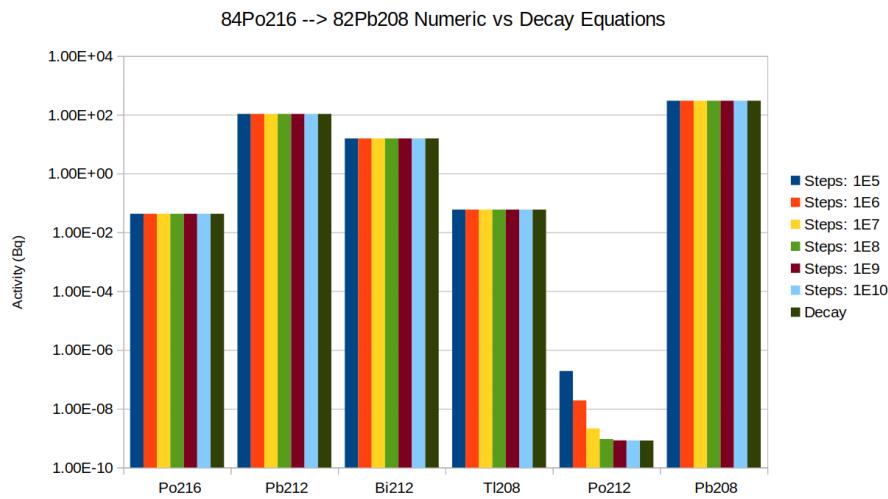


Figure 7.2: Decay path of Polonium-216

As seen in figure 7.2, for all but the shortest half life isotope Po-212, the numeric calculations are in good agreement with the decay equations. As the number of steps increases, and hence the time step decreases, the numerically computed activity for Po-212 also converges to that calculated with the decay equations.

The modified Bateman equation was also tested with several other decay chains against a numeric solution for each chain. These were the decay of  $^{49}Cr$ ,  $^{66}Ni$ ,  $^{125}Cs$ ,  $^{213}Bi$ ,  $^{216}Po$  and  $^{218}Rn$ . Apart from a break down in the numeric code, where the half life was comparable to or smaller than the time step, the results from both the equations and the numeric solver were in good agreement. The full results to this testing are included in appendix ??.

## 7.2 Experimental Proton Irradiation of Iron Target

### 7.2.1 Predicted Depth of Beam

In this work Iron has been irradiated with a proton beam from the University of Birmingham cyclotron. Early on it was determined that  $^{55}Co$  would be responsible for a 931KeV gamma. Preliminary investigations with SRIM into the stopping distance of protons in a range of materials shows 36MeV ions stop within 3mm in Iron and Steel(figure 7.3).

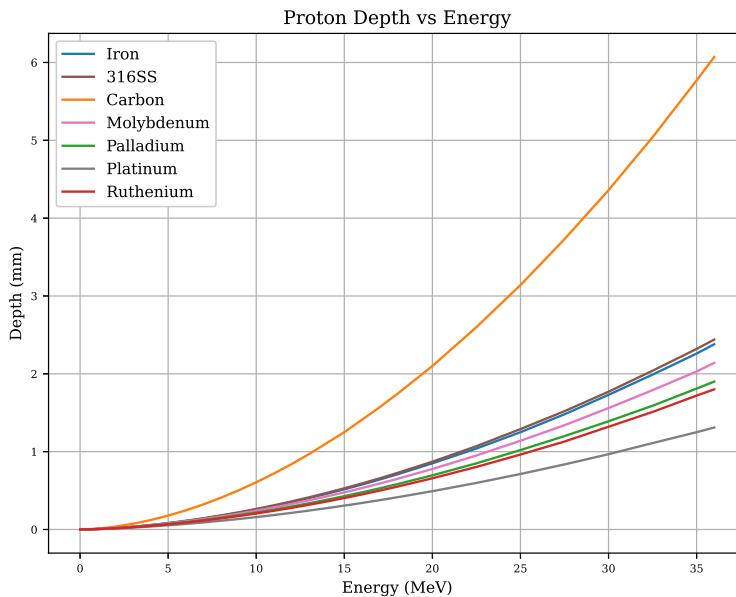


Figure 7.3: Proton stopping distances in various materials

It is clear that 3mm thick samples of Iron or Steel will completely stop 36MeV protons and 1mm of the same is enough to stop any with an energy of 20MeV or less.

### 7.2.2 Creation of Cobalt-55

$^{55}Co$  is a relatively short lived isotope with a half life of 17.5 hours and this is a concern to health during and shortly after irradiation. Isotopes with a long half life are not as active in small quantities, and those with very short half life decay away quickly.  $^{55}Co$  decays into another radioactive isotope,  $^{55}Fe$ , which then decays into  $^{55}Mn$  (figure 7.4). The second step in the decay chain is 2.73 years, so the first step with a much shorter half life will be responsible for the majority of the radiation shortly after being irradiated by protons.

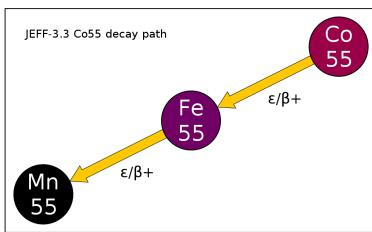


Figure 7.4: Decay path of Cobalt-55

The decay of  $^{55}\text{Co}$  will produce a range of gammas (figure 7.5), but one in particular to focus on when measuring the radioactivity is the 931KeV gamma with an intensity of 75%.

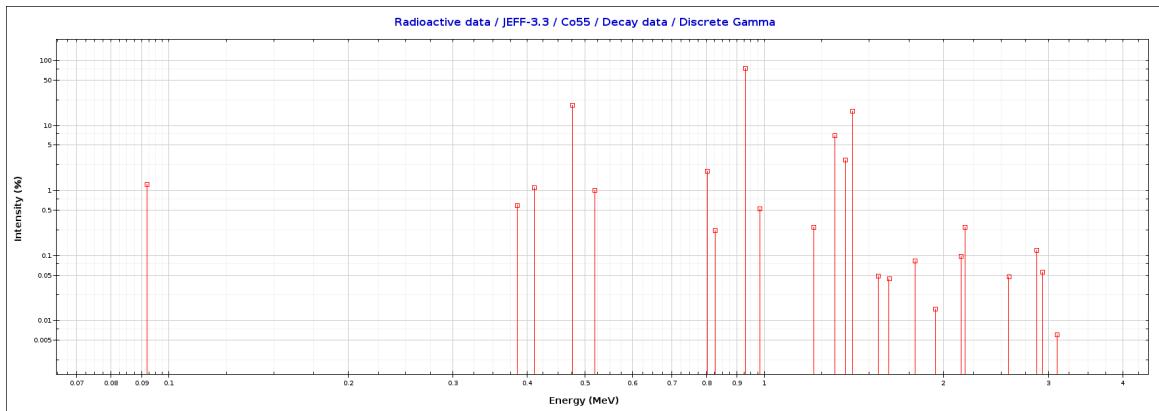


Figure 7.5: Cobalt-55 Gamma Intensity Plot

The stable isotope  $^{54}\text{Fe}$  makes up almost 6% of natural iron, and one reaction possibility with a proton results in  $^{55}\text{Co}$  and a gamma,  $^{26}\text{Fe}(p, \gamma)^{55}\text{Co}$ . The most abundant isotope in natural Iron also has a route to transmute to  $^{55}\text{Co}$  through the capture of a proton and the loss of two neutrons,  $^{26}\text{Fe}(p, 2n)^{55}\text{Co}$ .

Due to the short half-life and intensity of the 931KeV gamma, this peak was measured in particular after irradiating the iron target. It was also estimated manually using the TENDL-2009 and TENDL-2019 cross section databases.

### 7.2.3 Irradiation Parameters and Resulting Gamma Count

A 0.5mm thick pure iron target was irradiated for five minutes at a current of 0.5 microamps with 36MeV protons. The proton beam formed a  $64\text{mm}^2$  square that was taken from the cyclotron through one of the beam lines, with the target perpendicular to the beam.

The target was left to cool for several days before being safe enough to handle. The radioactivity was measured using a high purity germanium detector which was calibrated to detect approximately 4 out of every 100 gammas emitted.

- 1 RANGE: 617 = 911.51keV to 649 = 958.79keV
- 2 AREA : Gross = 2127168 Net = 1342565 +/- 2207
- 3 CENTROID: 631.74 = 933.29keV
- 4 SHAPE: FWHM = 6.82 FW(1/5)M = 10.06
- 5 ID: Bi-214 at 934.05keV
- 6 Corrected Rate = 35349.26 +/- 58.11 cA

Listing 7.1: Maestro 931KeV Peak Measurement

The gamma counts were recorded over a range of 0MeV to 3MeV (figure 7.6). Correcting the measured count for the 931KeV peak, to account for the geometry and detector, the count rate for 931KeV gamma rays from  $^{55}_{27}Co$  was measured at  $4.43 \times 10^4 + / - 1.05 \times 10^3$  counts per second.

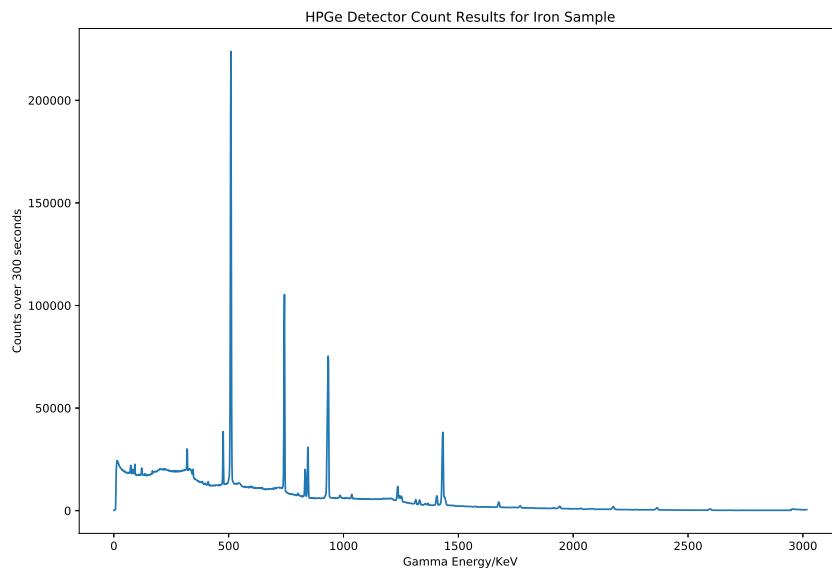


Figure 7.6: Gamma counts from the irradiated iron target over 300 seconds

There are several decay modes that contribute to the radioactivity of the sample, and the largest contributors to gamma emission also undergo either electron capture or  $\beta^+$  emission. The decay paths are:

- $^{55}_{27}Co \xrightarrow{\epsilon/\beta^+} {}^{55}_{26}Fe \xrightarrow{\epsilon/\beta^+} {}^{55}_{25}Mn$
- $^{56}_{27}Co \xrightarrow{\epsilon/\beta^+} {}^{56}_{26}Fe$
- $^{54}_{25}Mn \xrightarrow{\epsilon/\beta^+} {}^{54}_{24}Cr$
- $^{52}_{25}Mn \xrightarrow{\epsilon/\beta^+} {}^{52}_{24}Cr$
- $^{52}_{25}Mn^* \xrightarrow{\epsilon/\beta^+} {}^{52}_{24}Cr$

Where positrons are emitted, they will annihilate with an electron. This event will release two photons in opposite directions, each of which will have a 511KeV energy. This is responsible for the 511KeV peak measured in figure 7.6. However, at the time of writing, the computer code will not compute this peak and it will be omitted from the computational results.

## 7.3 Simulating Proton Irradiated Iron with SRIM

The trajectory data used by the activity code(s), and also to estimate the  $^{55}_{27}Co$  activity, was provided by SRIM. An input configuration was set with Iron as the target material and protons as the projectile ion. The sample depth was set to at least the target thickness, 0.5mm. The activity code truncates the data to fit the target thickness, so a much larger depth, say 1m, could be set in SRIM to ensure all the ions stop within the simulated target, and this data file may be reused for targets of varying thicknesses.

The ions travelling through the thin 0.5mm target lose between 4MeV and 5MeV (figure 7.7b) as they pass, and they do so quite smoothly (figure 7.7c). The majority of the energy is lost smoothly through electronic stopping.

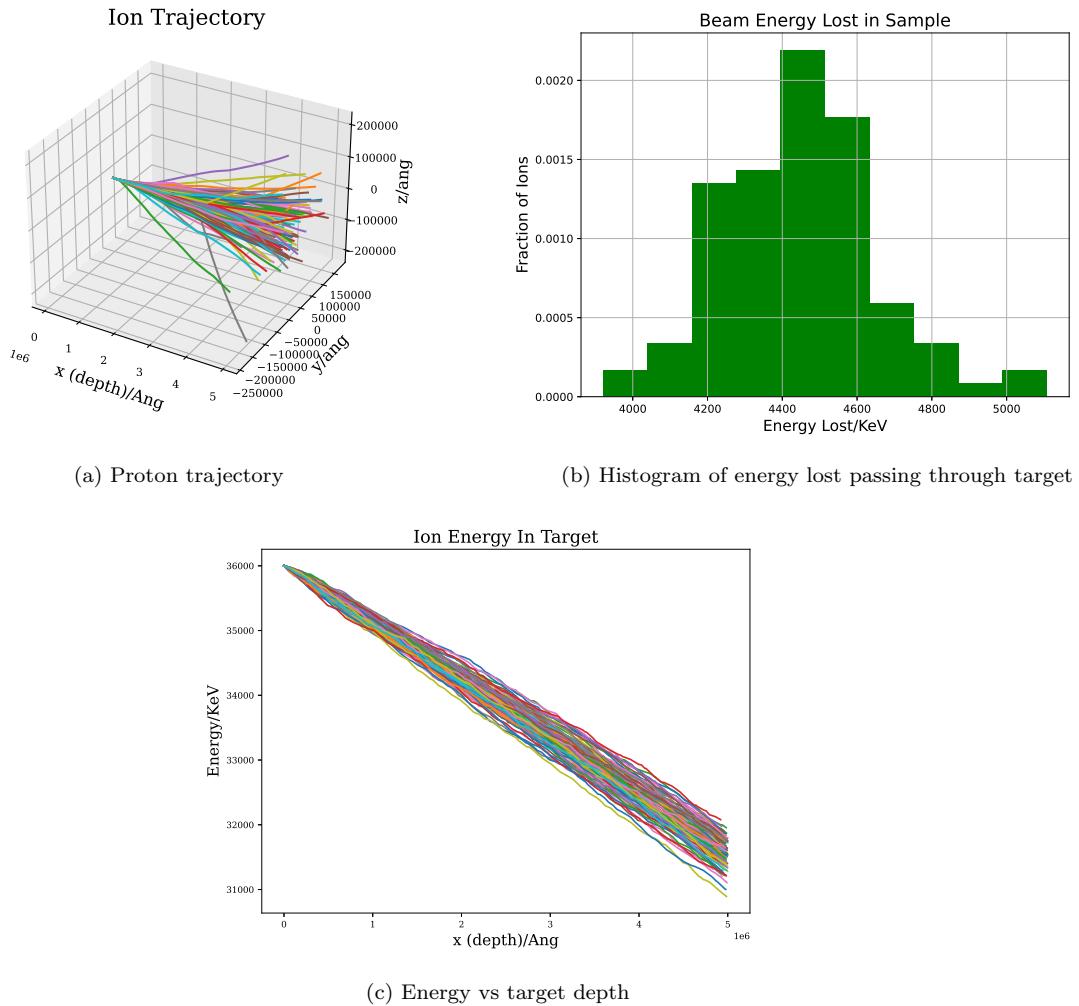


Figure 7.7: Protons through a 0.5mm thick iron target - 100 proton histories

The ion trajectory plot shows the deflection of ions in both the y and z axis. A large percentage of the ions here deviate very little from a straight pass through, but several are deflected by quite a large angle. Within the activity computer codes, these ions deflected ions will travel a greater distance through the target. The only cutoffs applied to the ion trajectory data are either the ion reaching 0MeV or travelling more than the depth in the x-axis. Any ions travelling a large distance in the y or z axis are still determined to be within the target.

As the thickness of the iron is increased, all the energy is lost to the iron target, and this loss of energy occurs within a shorter range in the remaining quarter of a millimetre depth of the iron target (figure 7.8a).

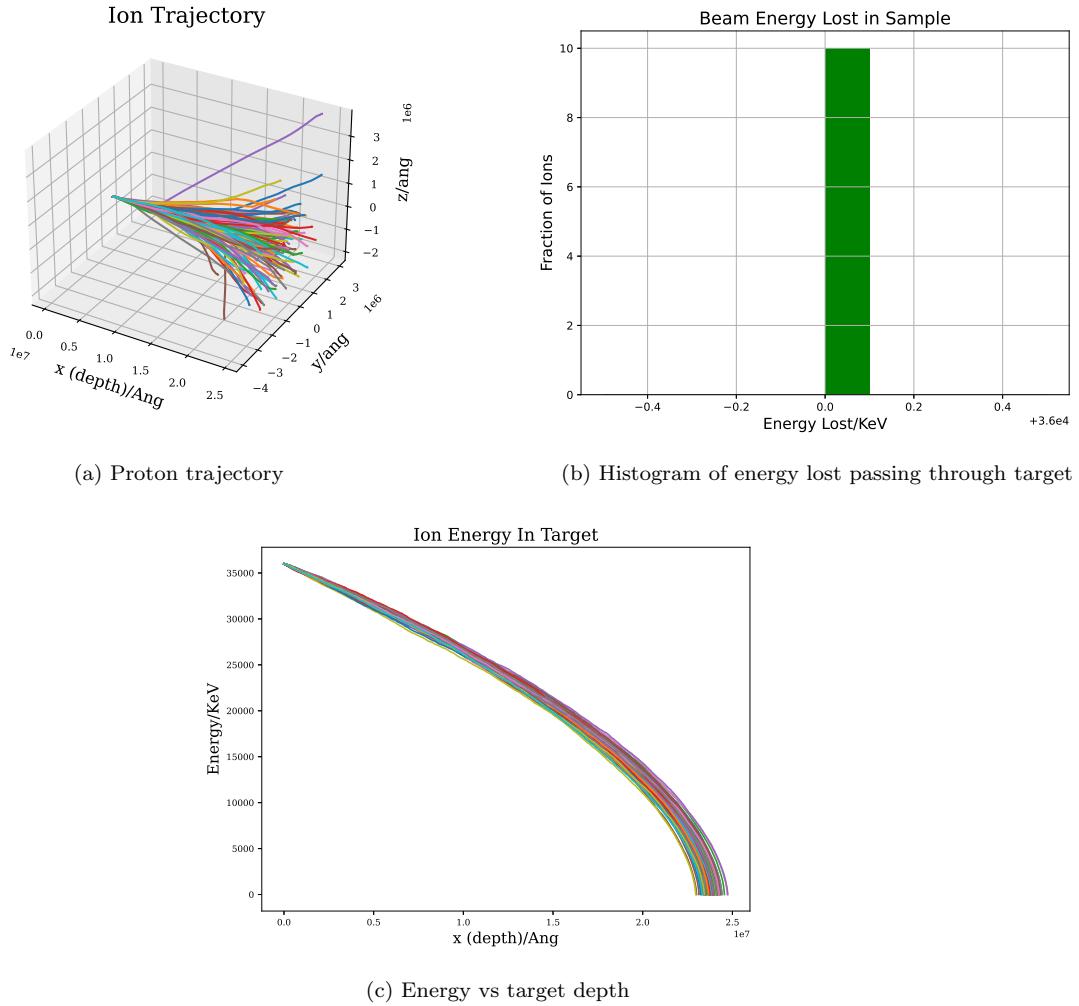


Figure 7.8: Protons through target thick enough to stop all ions - 100 proton histories

## 7.4 Estimation of Cobalt-55 Radioactivity

There are over ten TENDL libraries available, and the available data and the range of data points is inconsistent across these different versions. The earlier library, that was used in the first version of the Activity code, covers a range up to 200MeV for Iron-proton reaction cross sections. Many of the more recent data files contain total reaction cross sections only, and in the most recent data files, 2017 and 2019, the energy range only covers proton reactions from 0MeV to 28MeV, abruptly dropping to 0 barns at 30MeV as seen in fig 7.9.

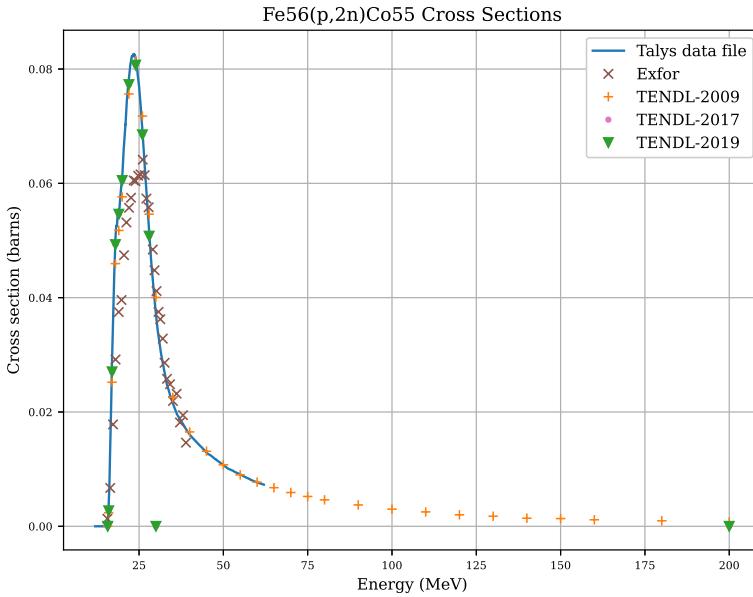


Figure 7.9:  $Fe^{56}(p, 2n)Co^{55}$  cross section data from several sources [170][171][172][173][174]

The activity of the sample, in particular due to the radioactivity of  $^{55}Co$ , was estimated using a number of data sources. As a consequence of the short beam duration when compared to the half-life of Cobalt-55, any loss during irradiation is ignored in each case.

The estimations use one set of experimental data, from the EXFOR data file, as well as evaluated and calculated data points. As may be seen in 7.9, all data sets follow a similar path although the peak is slightly lower for the experimental data. The two most recent TENDL data plots terminate abruptly, but the TENDL-2009 and TALYS calculated data points continue in agreement with one another up to 62MeV, where the calculated points end.

The parameters in relation to the beam current, duration, target material and the gamma intensity were consistent between the estimations. They are the same parameters as used in the experimental work and are given in table 7.3.

Parameter	Value
ND/atoms $^{56}Fe$	$7.4 \times 10^{28}$
ND/atoms $^{54}Fe$	$4.6 \times 10^{27}$
J/amps	$0.5 \times 10^{-7}$
Q/C	$1.60 \times 10^{-19}$
d/m	$5.0 \times 10^{-4}$
irradiation time/s	$3.0 \times 10^2$
Co55 t-half/s	$6.3 \times 10^4$
Co55 $\lambda$	$1.1 \times 10^{-5}$
931KeV Gamma Intensity	0.75

Table 7.3: Parameters used to estimate Co55 931KeV gamma activity

The 2019 data file ranges up to 30MeV for the  $^{54}Fe(p, \gamma)^{55}Co$  and  $^{56}Fe(p, 2n)^{55}Co$  reactions. As the data cuts off at 30MeV, the cross section data for a 28-30MeV proton is used in the estimation. This results in an over estimate in activity.

A 36MeV proton is expected to lose up to 4-5MeV travelling through a 0.5mm thick Iron target, as seen in figure 7.7c. However, in this estimation the ions are assumed to be only 36MeV (or 28MeV, where 36MeV data

are unavailable).

	TENDL 2009	TENDL 2017	TENDL 2019	Exfor	TALYS file	TALYS DB
$^{54}_{26}Fe(p, \gamma)^{55}_{27}Co$						
P energy (MeV)	36	28	28	36	36	36
XS $\sigma$ (barns)	$1.29 \times 10^{-4}$	$1.94 \times 10^{-4}$	$1.94 \times 10^{-4}$	-	$1.33 \times 10^{-4}$	$1.75 \times 10^{-4}$
Reaction Rate	$9.32 \times 10^4$	$1.40 \times 10^5$	$1.40 \times 10^5$	-	$9.64 \times 10^4$	$1.27 \times 10^5$
Activity (Bq)	$1.78 \times 10^1$	$2.69 \times 10^1$	$2.69 \times 10^1$	-	$1.84 \times 10^1$	$2.43 \times 10^1$
Gamma (Bq)	$1.34 \times 10^1$	$2.02 \times 10^1$	$2.02 \times 10^1$	-	$1.38 \times 10^1$	$1.82 \times 10^1$
$^{56}_{26}Fe(p, 2n)^{55}_{27}Co$						
P energy (MeV)	36	28	28	36	36	36
XS $\sigma$ (barns)	$2.13 \times 10^{-2}$	$5.08 \times 10^{-2}$	$5.08 \times 10^{-2}$	$2.30 \times 10^{-2}$	$2.00 \times 10^{-2}$	$2.00 \times 10^{-2}$
Reaction Rate	$2.44 \times 10^8$	$5.81 \times 10^8$	$5.81 \times 10^8$	$2.64 \times 10^8$	$2.29 \times 10^8$	$2.29 \times 10^8$
Activity (Bq)	$4.67 \times 10^4$	$1.11 \times 10^5$	$1.11 \times 10^5$	$5.04 \times 10^4$	$4.38 \times 10^4$	$4.38 \times 10^4$
Gamma (Bq)	$3.50 \times 10^4$	$8.33 \times 10^4$	$8.33 \times 10^4$	$3.78 \times 10^4$	$3.29 \times 10^4$	$3.29 \times 10^4$

Table 7.4: Estimation of the Cobalt-55 931KeV gamma activity based on the TENDL-2009[170], TENDL-2017[171], TENDL-2019[172] data files, EXFOR experimental data [173] and TALYS generated data files[174].

The largest contribution to the 931KeV gamma is due to the  $^{56}_{26}Fe(p, 2n)^{55}_{27}Co$  reaction, with the  $^{54}_{26}Fe(p, \gamma)^{55}_{27}Co$  reaction being responsible for less than 0.1% of the gammas.

Out of the estimates calculated, the outliers are those that used data files with a 28 MeV maximum limit, and this is as to be expected. The final three results are much closer to one another, ranging from 43,800Bq to 50,400Bq for the overall activity and 32,900Bq to 37,800Bq for the rate of emission of Gammas. The full details including cross sections and reaction rates used are given in table 7.4.

## 7.5 Activity V1

A large proportion of the results for this part of the work was the development of a computer program to implement the decay equations and calculate reaction rates of isotopes, protons and production rates of residual isotopes.

### 7.5.1 Computer Code

Activity V1 uses the TENDL-2013 cross section database, as well as several other data files that contain element, isotope and decay data. This data has been processed and is packaged with the computer code. The source code and instructions on how to use the program are available to download from GitHub:

<https://github.com/BenPalmer1983/activity>

The user provides an ion trajectory file and an input file that specifies beam parameters and target composition. The program produced a number of plots including activity over time and expected gamma lines. The activity code with the derived activity equations was used to calculate the predicted radioactivity (figure 7.10) and predicted gamma lines (figure 7.11). A full output file and details of all isotopes and their activities are in Appendix ??.

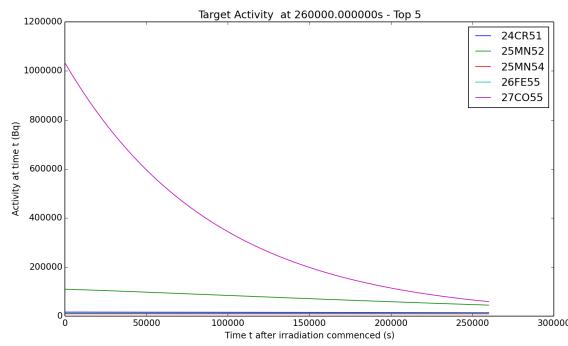


Figure 7.10: Isotopes with the top 5 radioactivity plotted over approximately 3 days

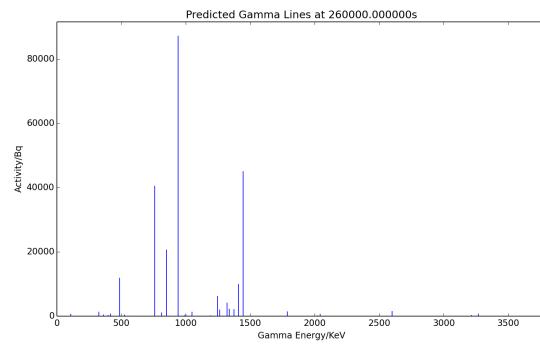


Figure 7.11: Predicted gamma lines approximately 3 days after irradiation

The code predicts a final 931KeV gamma activity of over 59,000 Bq and highlights several other isotopes of interest, including  $^{52}_{25}Mn$ ,  $^{51}_{24}Cr$ ,  $^{55}_{26}Fe$  and  $^{54}_{25}Mn$ , all of which have an activity of approximately 10,000 Bq or above. After 3 days of cooling, the target is predicted to dose an 80KG human, 1 meter away from the target, with  $4.0 \times 10^{-11}$  Gy/s (absorbed dose) which is less than the limit for a member of the public ( $6.34 \times 10^{-10}$  Si/s, effective dose).

## 7.6 Activity V2

There were a number of issues with version 1 of the computer code. It was written in Fortran, but over time this has been a stumbling block for new users, especially those new to Linux and unfamiliar with Fortran.

In the first version of the code, the trajectory data was replaced with a series of polynomials, but this method was not a reliable solution and it was preferred to use the data directly to calculate the reaction rates. Finally, a language such as Python3 is more widely used and contains very useful features such as dictionaries and a plotting module (matplotlib).

### 7.6.1 Computer Code Development

Activity V2 uses a custom cross section database generated using the TALYS nuclear reaction code along with the JEFF3.1.1 datafile for decay data. All the data has been processed and is packaged with the source code. The source code and instructions on how to use the program are available to download from GitHub:

[https://github.com/BenPalmer1983/activity\\_v2](https://github.com/BenPalmer1983/activity_v2)

As with version 1, the user provides an ion trajectory file and an input file that specifies beam parameters and target composition. The code may be used for both protons and deuterons, providing the correct trajectory file has been generated. It does also contain neutron reaction cross sections and may be used for very thin foils with a flat distribution of neutron energies.

### 7.6.2 Taly's Generated Proton Cross Section Database

A set of proton reaction cross section data files already exist and are updated every few years. The type and range of data changes between versions of the file. For example, the TENDL-2014 version contains only elastic and (z,anything) files for (p,Fe56). The TENDL-2009 file contains a wider ranger of reactions and under 50 data points span the energy range from 0MeV to 200MeV. The 2019 version contains even more reactions but spans a much shorter energy range, 0MeV to 28MeV, with cross sections given at just under 60 energy points.

The cyclotron used in the experimental work has an energy range up to 40MeV for protons and 53MeV for  ${}^4He^{2+}$  ions. To cover this entire range, the TALYS code was used to compute cross section reaction data from 0.01MeV to 62.0MeV in increments of 10KeV, breaking the energy range up into 6,200 data points for each reaction.

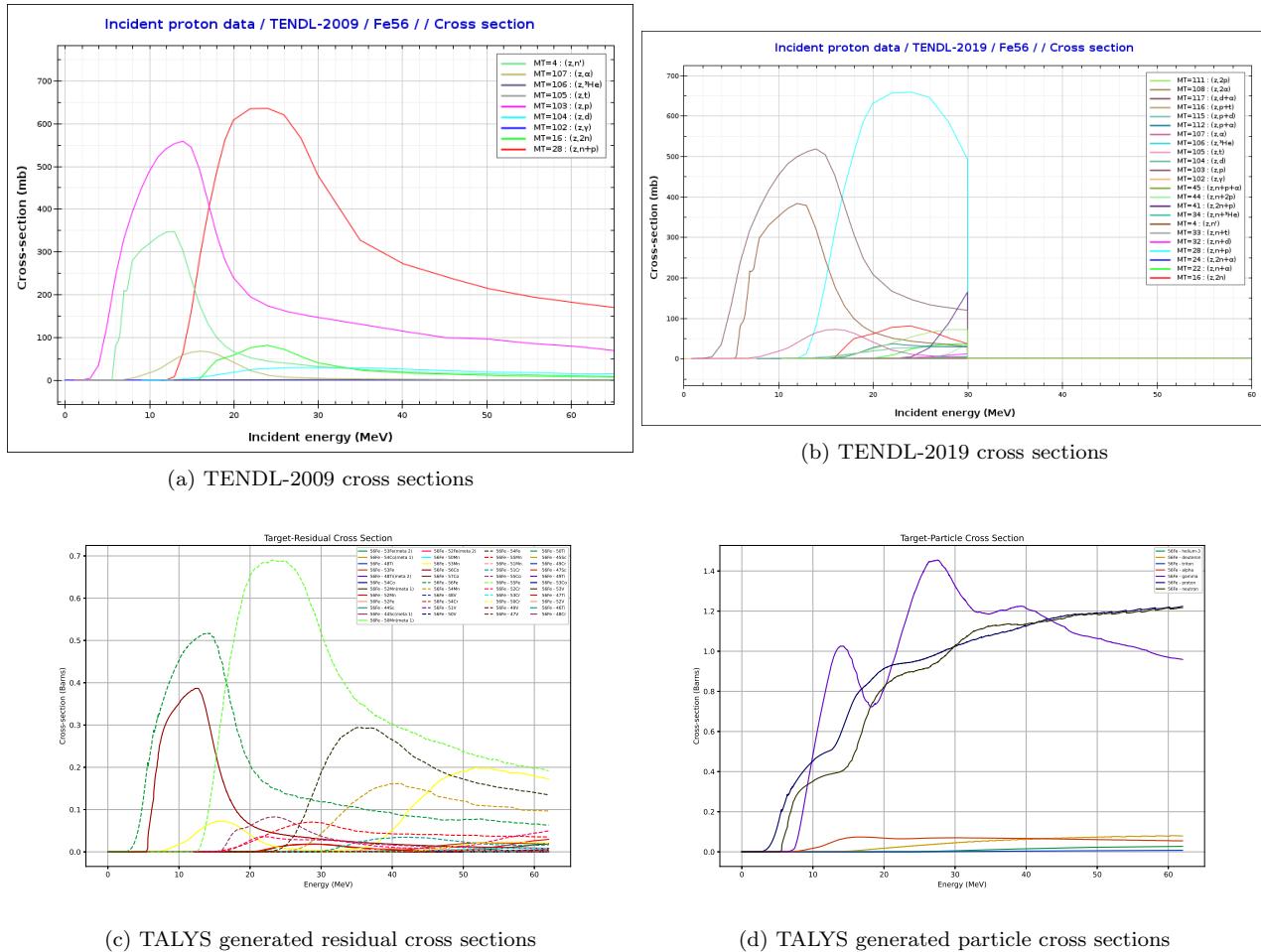


Figure 7.12: Cross sections for  $({}^{56}_{26}Fe, p)$  reactions

Due to limitations of the TALYS code, the calculations were processed in batches of either 400 or 600 points. The data was collated and processed resulting in a datafile of particle production reactions and residual production reactions. This is accessible through a stand-alone python class that is also a part of the Activity V2 code.

The residual reactions are grouped by target isotope and then by residual isotope, whereas the particle cross sections are grouped by target isotope then particle type. The computed values are plotted for all available reactions and all residuals for  ${}^{56}_{26}Fe$  reactions with protons (figure 7.12) and these show a good agreement, although there is a more detailed set of residual isotopes for the generated cross sections.

Listing 7.2: Status report from TALYS database code

```

1 #####
2      TALYS Cross Section Python Dictionary
3 #####
4
5      DB STATUS
6
7 Talys dir:      ../../data/talys
8 Isotopes file:  ../../data/isotopes.pz
9 Database load time: 24.489s
10
11 Projectiles: n d p

```

---

```

12
13 Total data points: 18777965
14
15 End time: 24.674s

```

---

The complete set of data files, and Python class used to read them, are available to download at:

[https://github.com/BenPalmer1983/talys\\_xs\\_db](https://github.com/BenPalmer1983/talys_xs_db)

The cross section data output by TALYS was grouped into four main files: elastic\_xs.pz, nonelastic\_xs.pz, particle\_xs.pz and residual\_xs.pz. All data files that make up this database currently total over 400MB and contain over 18 million data points (listing 7.2), although this could increase if more projectiles and a wide range of energies are added.

### 7.6.3 Simulated Proton Irradiated Iron

The iron irradiation experiment was modelled using the second version of the activity code. In this version the entire cross section database was created using the TALYS computer code (7.6.2). The same decay equations used in version 1 were also used here.

As the iron sample is irradiated a large number of particles (neutrons, protons, deuterons, tritons, helium-3, alphas and gammas) are created. Whilst the beam is on, these are produced at a rate of more than  $10^9$  per second (figure 7.13).

As irradiation commences, the isotope of iron responsible for most of the non-particle radiation is  $^{54}_{26}Fe$ , despite having a natural abundance of only 5.8%. Two residuals in particular resulting from proton reactions with  $^{56}_{26}Fe$  are responsible for the activity and these are  $^{54}_{25}Co$  and  $^{53}_{26}Fe$ .

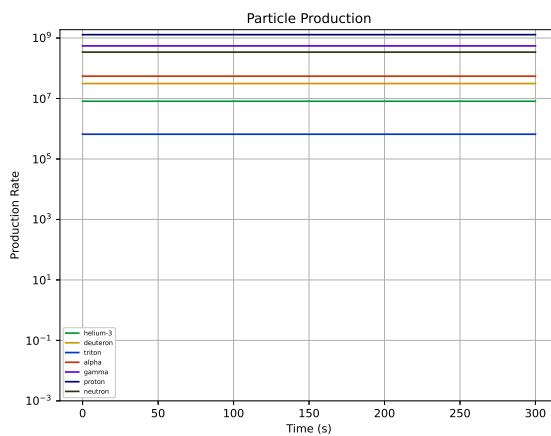


Figure 7.13: Particle production rate in beam

With a short half life of 0.19 seconds,  $^{54}_{25}Co$  is responsible for the majority of decays per second within a few seconds of irradiation starting. However, at this same time the majority of the gamma energy from residual isotopes originates from the decays of  $^{53}_{26}Fe$  (meta 1) and  $^{52}_{25}Mn$  (meta 1). The average gamma energy per decay of these isotopes is 3.03MeV and 1.42MeV respectively, and this is the reason for such a high contribution towards the gamma energy output (7.14a).

By the time the beam is turned off, at 300s, the primary isotope contributing almost 50% of the decays per second is  $^{53}_{26}Fe$ . It's contribution to the over all gamma output is still quite low with the main contributor being  $^{52}_{25}Mn$  (meta 1) as was the case at the start of irradiation (7.14b). This difference between the decays and gamma energy is due to  $^{53}_{26}Fe$  having an average gamma energy of only 0.18MeV per decay, whereas the metastable  $^{52}_{25}Mn$  has a much higher average gamma energy per decay of 1.4MeV.

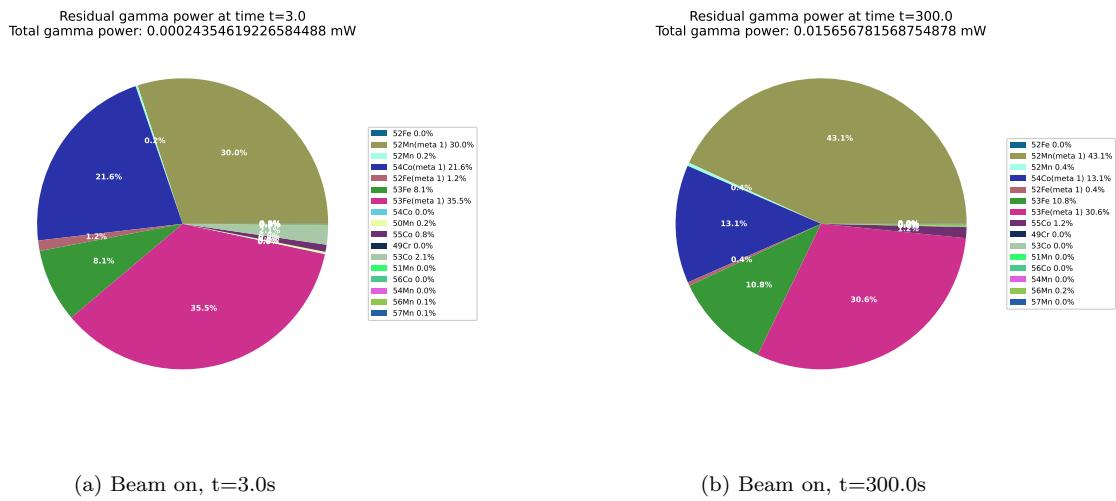


Figure 7.14: Gamma output of residual isotopes, by isotope, at the start and end of irradiation

Once the beam is turned off, those isotopes with a very short half life decay away quickly. After 1 day of cooling, the isotopes responsible for activity in the sample are very different. With a half life of less than 10 minutes, the  $^{53}Fe$  will have decayed to nothing as would have the  $^{53}Fe$  (meta 1) and  $^{52}Mn$  (meta 1). At this time, the isotope with the highest contribution to decays per second will be  $^{55}Co$ , but with a half-life of 17.53 hours it does not remain the largest contributor. It is also responsible for more than half of the gamma power output at this time as well (7.15a).

At the end of the experiment, the time at which the experimental gamma readings were taken ( $2.6 \times 10^5$ s), the largest contributor to both decays per second and gamma energy output per second is the  $^{52}Mn$  isotope (7.15b).

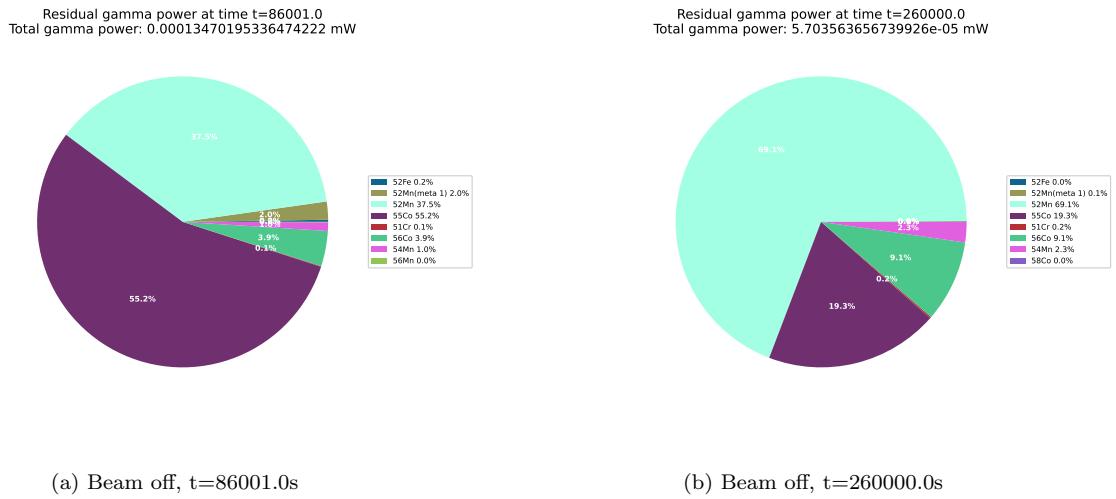


Figure 7.15: Residual isotope power output, by residual, once beam is off.

The computer code also calculates the predicted gamma lines as time goes by. The gamma lines become more prominent whilst the ion beam is irradiating the target to a maximum activity at 300 seconds. The three most active gammas at this point are computed to be 1.43MeV, 378KeV and 701KeV. The 1.43MeV gamma is released by the decay of the metastable isotope  $^{52}Mn$  (meta 1) and is emitted at a rate of  $2.9 \times 10^7$ Bq. The 377KeV and 791KeV gammas are emitted by  $^{53}Fe$  and  $^{53}Fe$  (meta 1) respectively having emission rates of  $2.4 \times 10^7$ Bq and  $9.7 \times 10^6$ Bq.

As noted previously, the isotopes responsible for change over time and in particular when the source, the proton beam, is removed. Following a day of cooling the 931KeV gamma from  $^{55}Co$  will have an activity of

approximately  $2.9 \times 10^5$ Bq and this is accompanied by another gamma of similar energy. This is the 935KeV gamma from the decay of  $^{52}_{25}Mn$ . At this point in time the activity of the 935KeV gamma is less than that of the 931KeV gamma with an activity of  $9.49 \times 10^4$ Bq. In addition, the  $^{52}_{25}Mn$  will also emit a 1.43 MeV gamma at a similar rate calculated to be  $1.0 \times 10^5$ Bq. With half lives of 17.5 hours for  $^{55}_{27}Co$  and 134 hours for  $^{52}_{25}Mn$  it is clear that, at some point, the activity of the 931KeV gamma will drop below that of the 935KeV gamma.

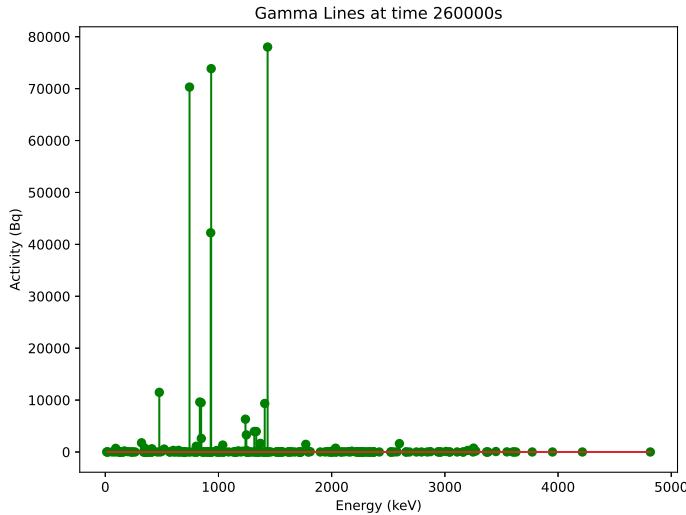


Figure 7.16: Expected gamma lines by the end of the experiment (3 days)

At the end time of the simulated experiment,  $2.6 \times 10^5$ s, the activity of the 931KeV Gamma is predicted to have dropped to just over 42,000Bq. It's neighbour, the 935KeV Gamma, will have an activity almost double, with 74,000 decays per second (figure 7.16). The full results calculated by this code are included in the appendix, chapter ??.

The prediction of such a large neighbouring Gamma does raise a question where the experimental value is concerned. It may be that, due to the resolution of the Gamma detector, the measured peak is the sum of both the 931KeV and 935KeV Gamma. Ideally the detector would be calibrated for a smaller range about that peak and the experiment would be repeated.

#### 7.6.4 Comparing Experimental Results with Calculated Values

The estimation is cruder and does not take into account the details of the loss of energy of the projectile as it passes through the target. It also does not take into account the creation of  $^{55}_{27}Co$  through other reactions. It gives a lower value of activity than the computer codes and the experimentally measured value (table 7.5).

Value source	Co-55 Gammas/second at approximately 3 days
Experimental	$4.4 \times 10^4 + / - 1.05 \times 10^3$
TENDL-2009	$3.5 \times 10^4$
Activity V1	$5.9 \times 10^4$
Activity V2	$4.2 \times 10^4$

Table 7.5: Comparison of proton irradiated iron Co55 931KeV gamma rates

## 7.7 Predicted Activity for 100DPA Irradiated Iron

Generation IV reactors are required to withstand 100-150DPA over the lifetime of the plant, which may span 30 years or more. The Scanditronix cyclotron is capable of creating a 60 microamp beam and, due to the small area (and volume of material) this may be concentrated on, it is able to create high damage rates over a shorter period of time.

The Activity V2 code was used to calculate how radioactive a 100DPA iron sample would be after irradiation at the maximum current setting of 60 microamps. The simulated target is 0.5mm thick, pure iron and the beam is protons and has an area of  $64\text{mm}^2$ . The projectile fluence is  $3.75 \times 10^{14}$  protons per second, and the number of atoms within the volume of iron are  $2.56 \times 10^{21}$ . Five different energy settings were used (5MeV, 10MeV, 15MeV, 20MeV, 25MeV) and the vacancies per ion as well as the ion trajectories were generated by SRIM.

	5MeV	10MeV	15MeV	20MeV	25MeV	30MeV
VPI	38.5	58.8	51.0	23.7	20.2	12.8
DPA/year	178	272	236	109	93	59
Days for 100 DPA	205	134	155	333	391	617
Damage Depth (mm)	<0.1	0.25	0.5+	0.5+	0.5+	0.5+
Activity <sup>1</sup> (Bq)	$2.27 \times 10^6$	$7.32 \times 10^8$	$2.55 \times 10^9$	$3.66 \times 10^9$	$4.10 \times 10^9$	$4.88 \times 10^9$
Gamma Output <sup>1</sup> (mW)	$9.44 \times 10^{-5}$	$3.87 \times 10^{-1}$	$1.32 \times 10^0$	$1.29 \times 10^0$	$6.05 \times 10^{-1}$	$5.74 \times 10^{-1}$
Activity <sup>2</sup> (Bq)	$2.14 \times 10^6$	$6.86 \times 10^8$	$2.35 \times 10^9$	$3.08 \times 10^9$	$2.82 \times 10^9$	$3.52 \times 10^9$
Gamma Output <sup>2</sup> (mW)	$8.85 \times 10^{-5}$	$3.64 \times 10^{-1}$	$1.24 \times 10^0$	$1.16 \times 10^0$	$3.63 \times 10^{-1}$	$2.94 \times 10^{-1}$

Table 7.6: Comparison of proton irradiated iron up to 100DPA at a range of proton energies. The first set of measurements<sup>1</sup> are calculated values at the point the proton beam is turned off. The second set<sup>2</sup> are calculated after 1 week of cooling.

In each case the sample was irradiated close to saturation. Doubling the irradiation time would not be expected to significantly increase the activity, as sufficient time (in each case) is given for the radioactive isotopes being created to balance those decaying away.

It is clear to see from the predicted activities in table 7.6 that the higher energy beams would take longer to reach 100DPA, as they pass through the target leaving at a high energy, whilst also irradiating the target several orders of magnitude more than 5MeV and 10MeV.

The Coulomb potential between a proton and an iron nucleus, at the combined nuclear radius of both, is approximately 6.5MeV (eq. 4.2 and eq. 4.3). Unlike neutrons radiation, protons need to overcome the Coulomb repulsion and this explains the relatively low radioactive activation by the 5MeV beam.

The predominant isotope responsible for radiation in the 5MeV sample is expected to be  $^{58}\text{Co}$ , and with a half life of 77 days this will remain radioactive for a substantial amount of time (relative to testing by irradiating, cooling then testing the material).

For the 10-20MeV irradiated samples the primary isotope responsible for the radiation is  $^{56}\text{Co}$ , and its contribution to the total gamma output in terms of power ranges from 95% to 99%. This isotope has a similarly long half life, relative to the testing cycle, of 71 days. Depending upon the volume of material being irradiated, this could lead to problems whereby the sample may need to cool for months to reach a safe activity to handle.

As higher energy protons are used to damage the sample a wider range of reaction possibilities become more likely including that leading to both  $^{52}\text{Mn}$  and  $^{52}\text{Mn}(\text{meta}1)$ . By the end of irradiation a large percentage of radioactivity is due to these isotopes (28.6% for 25MeV protons and 42.8% for 30MeV protons). The half life of  $^{52}\text{Mn}$  and  $^{52}\text{Mn}(\text{meta}1)$  are short, being 5.6 days and 21 minutes respectively, and will decay away leaving  $^{56}\text{Co}$  as the main contributor to radiation once again.

Detailed plots resulting from this Activity V2 simulation are in appendix ??.

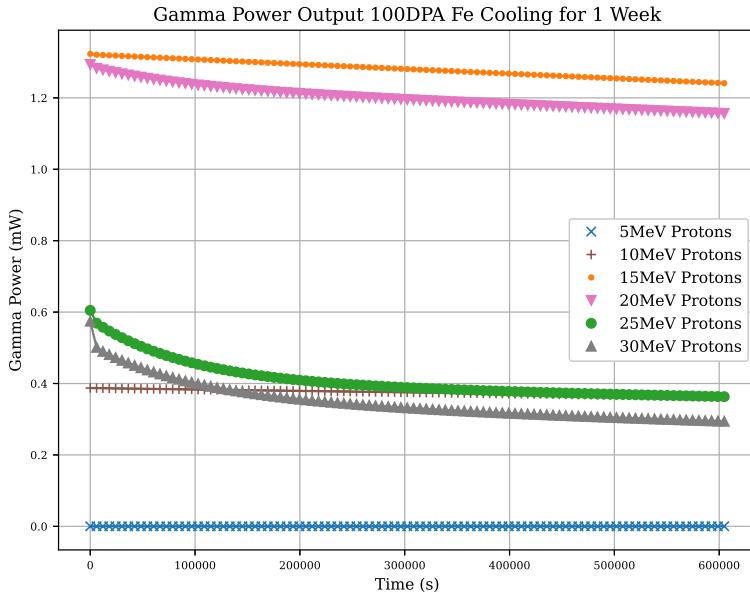


Figure 7.17: Change in isotope gamma output over time with proton beam off.

## 7.8 Proton Irradiated Molybdenum

A colleague at the University of Birmingham, John Hewett, irradiated several samples of Molybdenum for materials testing and throughout this the activity of the Molybdenum was measured by the department's HPGe detector[175]. Similar to the Iron sample, thin sheets of Molybdenum were irradiated using a beam line from the cyclotron.

### 7.8.1 Molybdenum Experimental Irradiation Results

A thin 49 micrometer sheet of pure Molybdenum was irradiated with a 13MeV proton beam for 19 seconds at a current of 0.5 microamps. The beam area on the target was  $50\text{mm}^2$ . A thick 506 micrometer sheet of the same was also irradiated, but this time a higher beam current of 5 microamps and a duration of 1,500s.

The detector was calibrated to detect energies in the range covering 0MeV to 1.2MeV and several measurements were taken for each sample. For the thick target the total gamma activity was measured as 16MBq at 3 days (14MBq for gammas above 200KeV) and 7.1MBq at 7 days (6.6MBq for gammas greater than 200KeV).

A full gamma spectra was also recorded for the thick sample by gamma peak at three days. The isotopes responsible for the five most prominent peaks at this point were  $^{95}_{43}\text{Tc}$  (half life of 20 hours[22]) and  $^{96}_{43}\text{Tc}$  (half life of 4.28 days[22]). The results measured and interpolated by my colleague are given in figure 7.19 and table 7.7.

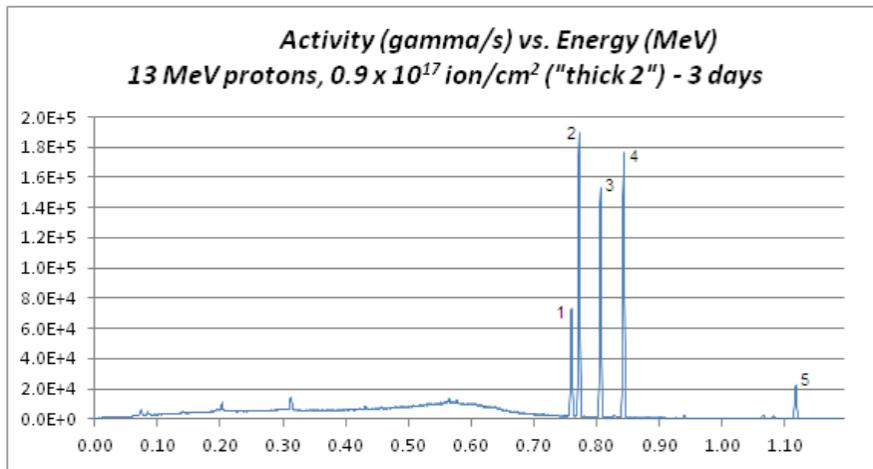


Figure 7.18: Proton irradiated Molybdenum gamma activity measured by colleague John Hewett

Peak	Energy (KeV)	Isotope	Gamma Intensity	Activity (Bq)	Measured Energy (KeV)
1	766	$^{95}_{43}Tc$	93.82%	$5.118 \times 10^5$	763
2	778	$^{96}_{43}Tc$	99.76%	$1.355 \times 10^6$	776
3	813	$^{96}_{43}Tc$	82.00%	$1.155 \times 10^6$	810
4	850	$^{96}_{43}Tc$	97.57%	$1.386 \times 10^6$	847
5	1127	$^{96}_{43}Tc$	15.16%	$2.103 \times 10^5$	1123

Table 7.7: Gamma emissions measured from a 0.5mm thick sample of Molybdenum irradiated with 13MeV protons at 5 microamps for 1,500 seconds

### 7.8.2 Estimating the Activity of Proton Irradiated Molybdenum

The gamma activity of the 0.5mm thick molybdenum target will be estimated. The generated data are compared to a number of experimental data sources through EXFOR.

Naturally occurring Molybdenum is made up of four stable isotopes, two observationally stable isotopes,  $^{92}_{42}Mo$  and  $^{98}_{42}Mo$  and one unstable isotope,  $^{100}_{42}Mo$ , with an extremely long half life of  $9.9 \times 10^{19}$  years (figure 7.19).

Molybdenum Natural Abundance of Stable Isotopes

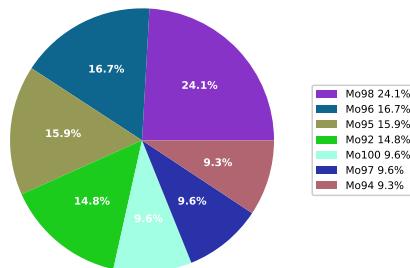


Figure 7.19: Isotopes in naturally occurring Molybdenum

There are five reactions of particular interest that are predicted to create radioactive isotopes at the highest rate. Two of the residual isotopes were identified in the experimental work and the reactions responsible are the first two of the five listed below.

- ${}_{42}^{95}\text{Mo}(p, n){}_{43}^{95}\text{Tc}$
- ${}_{42}^{96}\text{Mo}(p, n){}_{43}^{96}\text{Tc}$
- ${}_{42}^{96}\text{Mo}(p, n){}_{43}^{96}\text{Tc}$  (meta 1)
- ${}_{42}^{97}\text{Mo}(p, n){}_{43}^{97}\text{Tc}$
- ${}_{42}^{94}\text{Mo}(p, n){}_{43}^{94}\text{Tc}$

The isotope  ${}_{43}^{95}\text{Tc}$  produced four measurable gamma activities, and these gamma decay rates will be estimated using cross section data available from TENDL and EXFOR. The cross sections have been measured experimentally, with their results published, by four groups since the 1970s. Whilst there are discrepancies, the total cross section predicted by the TALYS code fits well with the TENDL-2019 database and measurements by Lamere[176] and Flynn[177]. The lower energy cross sections of Hogan[178] do not match as well but the higher energy reactions(30MeV+) are a closer match. Finally cross section measurements by Levkovski[179] are slightly larger near to the peak values, but on the whole match well with the TALYS computed values.

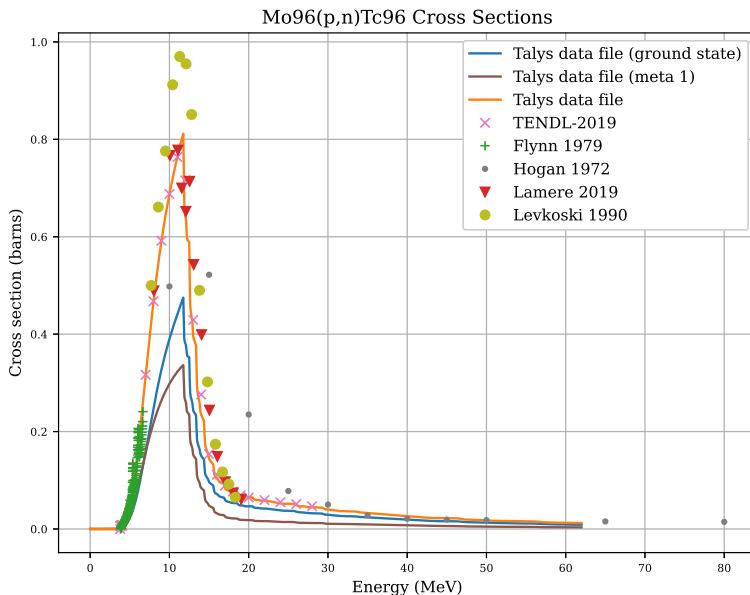


Figure 7.20: Reaction cross sections for  ${}_{42}^{96}\text{Mo}(p, n){}_{43}^{96}\text{Tc}$  from various sources

The cross sections computed by the TALYS code are split into both the reaction to the Technetium-96 ground state and metastable 1 state. The separate data are plotted in figure 7.20 along with their combined cross section.

This estimate will calculate the contribution from the ground state only. The metastable state of  ${}_{43}^{96}\text{Tc}$  has a 2% chance to decay directly to  ${}_{42}^{96}\text{Mo}$  and a 98% chance to decay to the ground state. The half life for the metastable state is 51.5 minutes[22] and, coupled with a lower reaction cross section than the ground state, it will contribute a relatively small amount of  ${}_{43}^{96}\text{Tc}$  when compared to those created by the ion beam and will delay the production of the ground state past the point at which the beam is removed.

The 13MeV protons will lose all their energy within the first 375 micrometers of the material as computed by SRIM (figure 7.3). As the ions lose energy the probability of a reaction occurring changes. The cross section ranges from a maximum of 475mb at 11MeV and this drops to zero once the ion energy falls to 3.8MeV. The average cross section over the entire energy range based on SRIM stopping distance calculations and TALYS data is 0.269 barns.

Given the data used is in good agreement with EXFOR experimental data, the 13MeV is expected to transmute  $2.93 \times 10^9$   $^{96}_{42}Mo$  atoms into  $^{96}_{43}Tc$  every second. By the end of the beam almost  $4.4 \times 10^{12}$  atoms will have been transmuted via this reaction and, as the half life of  $^{96}_{43}Tc$  is 247 times greater than the 1,500s beam duration, atoms lost to decay during irradiation will be ignored in this estimate. The intensity of each gamma varies, and these are listed in 7.7.

Peak	Energy (KeV)	Measured Activity (Bq)	Estimated Activity (Bq)	Factor of Over Estimate
2	778	$1.355 \times 10^6$	$4.74 \times 10^6$	3.5
3	813	$1.155 \times 10^6$	$4.15 \times 10^6$	3.6
4	850	$1.386 \times 10^6$	$4.93 \times 10^6$	3.6
5	1127	$2.103 \times 10^5$	$7.66 \times 10^5$	3.6

Table 7.8: A comparison between measured and estimated peaks from a 0.5mm thick sample of Molybdenum irradiated with 13MeV protons at 5 microamps for 1,500 seconds

There is a discrepancy between the measured amount and that predicted by past work, and it is consistent across the four gamma peaks. There are several simplifications with this estimate including the omission of the metastable state, but the inclusion of this would increase the estimated activity rather than decrease it.

My colleague investigated the affect of self attenuation of gammas passing through the Molybdenum sample. They made the assumption that the radioactive isotopes were concentrated half way through the thickness of the sample and that the gammas would need to pass through 250 micrometers of matter. It was expected that gammas above 200KeV would retain 94% of their intensity with those above 1MeV having 98.5% of their original intensity[175].

The distribution of radioactive isotopes would in fact be closer to the surface of the material than the midway point. This is illustrated in figure 7.21 and it shows the majority of the Technetium-96 is created in the top 150 microns. This plot is based on the SRIM stopping range for protons in Molybdenum and the cross section data used in the estimate.

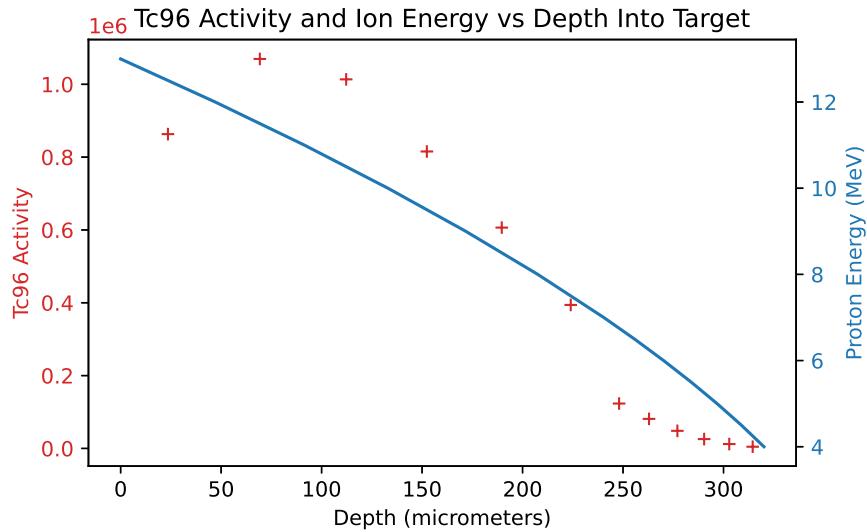


Figure 7.21: Technetium activity and beam energy as a function of depth into the target, showing a non-uniform distribution of radioactive atoms in the sample

It is plausible that the orientation of the sample would affect the gamma count. If it were placed such that the beam irradiated side faced the detector there would be less material between the radiation source and the detector. The opposite way round and more material would block the path of the gammas to the detector, thus decreasing their intensity.

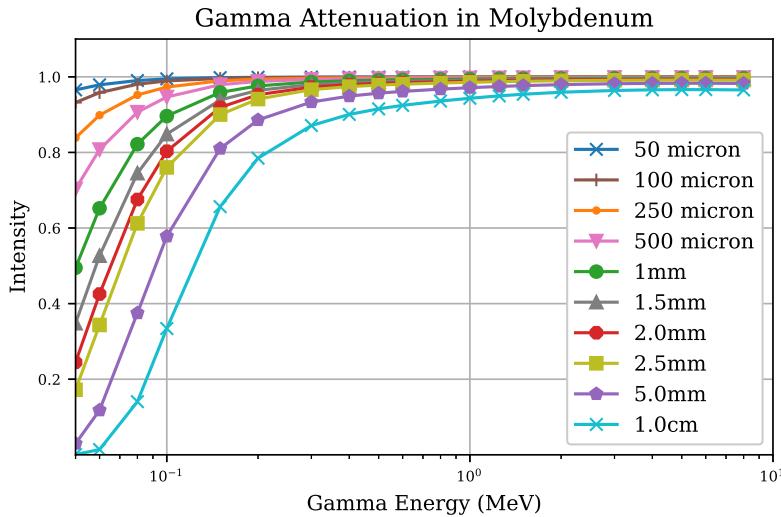


Figure 7.22: The attenuation of Gammas through Molybdenum for a range of energies and thicknesses[29]

When plotting the gamma attenuation through Molybdenum over a range of gamma energies it is clear that those with an energy of 500KeV or more drop to an intensity of as low as 0.9 in targets up to 1mm thick (figure 7.22)[29]. This is double the target depth of the thick target and so it does not explain the discrepancy between the experimental and estimated values.

### 7.8.3 Simulated Proton Irradiated Molybdenum

Both the thin and thick samples irradiated experimentally are simulated. However, detailed data are available for the thick sample only. A qualitative comparison of the thin sample will be made with the experimental results available.

Only stable isotopes are included in the cross section data file currently and, as 3 out of the 7 “stable” isotopes are either observationally stable, or have an incredibly long half life, their reactions are not included. The addition of these isotopes could highlight additional reactions but the data would need to be created first, if possible, using TALYS.

Out of the four stable isotopes of Molybdenum,  $^{96}\text{Mo}$  is responsible for most of the radioactivity both at the time the beam is turned off (1,500s) (figure ??) and after 3 days of cooling (figure ??). As discussed earlier this is due to the creation of both the ground state and metastable 1 state for Technetium-96. Whilst the beam is on over 60% of the activity is due to the metastable-1 isotope. The gamma energy emitted by the isomeric transition is comparatively low with an energy of just 32.28keV, and as a result it’s contribution to the power output is also low. In contrast the transition to the ground state for  $^{96}\text{Tc}$  (meta 1) releases many energetic gammas. The majority of the energy released towards the end of the ion irradiation is due to both the ground state and metastable 1 of Technetium-94 (figure ??).

After three days of cooling the almost 67% of the decays are due to the ground state Technetium-96 isotopes in the sample with another 31% from Technetium-95 (figure ??). The gamma energy output is also primarily due to these two isotopes, although with an 87% and 12% split (figure ??).

It is clear that there is a disagreement in the values of the experimental, estimated and computed activities and gamma lines. The fact that the estimated activities are lower than the calculated values was expected as the estimate only considered one reaction. The difference between the experimental work with both the estimated and calculated values is harder to explain but will be discussed in section ???. The full results from the activity code are included in appendix ??.

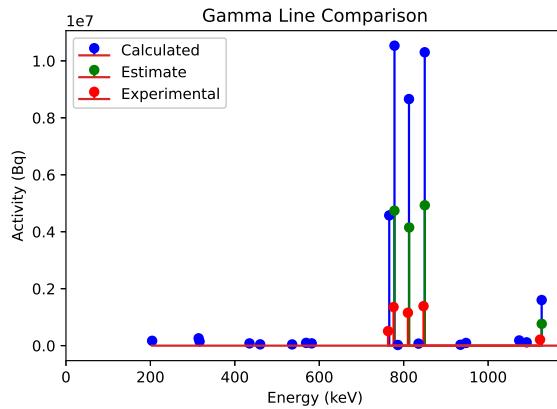


Figure 7.23: Comparison of Gamma peaks from experiment, a simple estimate and the Activity code.

Peak	Energy (KeV)	Measured Activity (Bq)	Computed Activity (Bq)	Factor of Over Estimate
1	766	$5.118 \times 10^5$	$4.574 \times 10^6$	8.9
2	778	$1.355 \times 10^6$	$1.053 \times 10^7$	7.8
3	813	$1.155 \times 10^6$	$8.658 \times 10^6$	7.5
4	850	$1.386 \times 10^6$	$1.030 \times 10^7$	7.4
5	1127	$2.103 \times 10^5$	$1.601 \times 10^6$	7.6

Table 7.9: A comparison between measured and calculated peaks from a 0.5mm thick sample of Molybdenum irradiated with 13MeV protons at 5 microamps for 1,500 seconds

## 7.9 Proton Irradiated Molybdenum Doped Steel

Another colleague in the Metallurgy and Materials department at the University of Birmingham, Alex Dickinson-Lomas, requested a copy of the Activity code and a simulation of a steel with a specific composition (Fe: 96.375, C: 0.772, Cu: 0.024, Mn: 1.36, Ni: 0.698, Si: 0.381, Cr: 0.092, V: 0.008, P: 0.009, S: 0.003, Mo: 0.278) that would be irradiated with 5MeV protons.

In comparison to the previous cases, 36MeV protons in Iron and 13MeV protons in Molybdenum, the activity is much lower. The simulated beam current was 0.5 microamps and the duration 5 minutes, but in contrast to the 100MBq maximum activity of the Iron sample, this is expected to reach a maximum of just over 10KBq with a gamma output of just  $1.0 \times 10^{-8}$ mW.

During irradiation, the isotope responsible for the majority of the residual radioactivity is  $^{13}N$  and this is created through the  $^{13}C(p, n)^{13}N$  reaction. This is despite the stable isotope carbon-13 having a percentage by mass in the sample of 0.008322%.

The reaction probability for  $^{13}C(p, n)^{13}N$  is also high in the 4-5MeV range with a cross section between 0.1 and 0.15 barns. In contrast the first reaction of note for Carbon-12 results in Boron-9 and this begins at proton energies near to and above 10MeV.

After a day of cooling the activity due to Nitrogen-13 will have decayed through 144 half lives to nothing. The reaction responsible for almost 40% the radioactivity at this point is predicted to be  $^{58}Fe(p, n)^{58}Co(meta1)$ . Other residual isotopes of note are  $^{57}Co$ ,  $^{58}Co$ ,  $^{55}Fe$  and  $^{55}Co$ . A full breakdown is included in appendix (??).

Whilst there are no experimental activity data to compare to, the Activity code is a useful tool. It takes a complex material and highlights likely reaction interactions, highly radioactive residual isotopes and gamma peaks. It also predicts the total activity expected and the calculation may be run multiple times with a range of target thicknesses and beam energies.

# Chapter 8

## Methodology: Interatomic Potential Fitting

*In order to perform MD ion damage simulations, an iron-palladium potential is needed. Experimental data is available for FCC palladium, but DFT is needed to compute the properties of FCC iron, as it does not exists in this phase at standard conditions without the addition of an austenite stabilizer. Additional DFT calculations will be used to compute the energy, stress and forces of slightly perturbed atoms which will in turn be used to fit the potentials. A new computer code will be created to fit the potentials.*

### 8.1 Interatomic Potential Fitting

The potentials in this work are fitted to both experimental data and DFT data. As the purpose of these potentials is to provide a way to model Palladium and Ruthenium, and determine whether or not they migrate to the surface, appropriate atomic configurations are included in the fitting process.

The work may be broken into three main parts. The first is devoted to using DFT to generate the required data, the second is the development of a computer program for fitting the potential, and the third is finally fitting the potential. An overview of this process is given in figure 8.1.

The process follows aspects of previous work. The force matching method of Ercolessi and Adams[121] and its application within Brommer's PotFit code[135] (and it's subsequent extension by Sheng[136]) impacted this work greatly. The exact fitting methods used by Bonny and spline type functions used throughout work on Iron alloys by Ackland, Mendelev, Hepburn and more helped to fix on a specific form of the functions and functional used here.

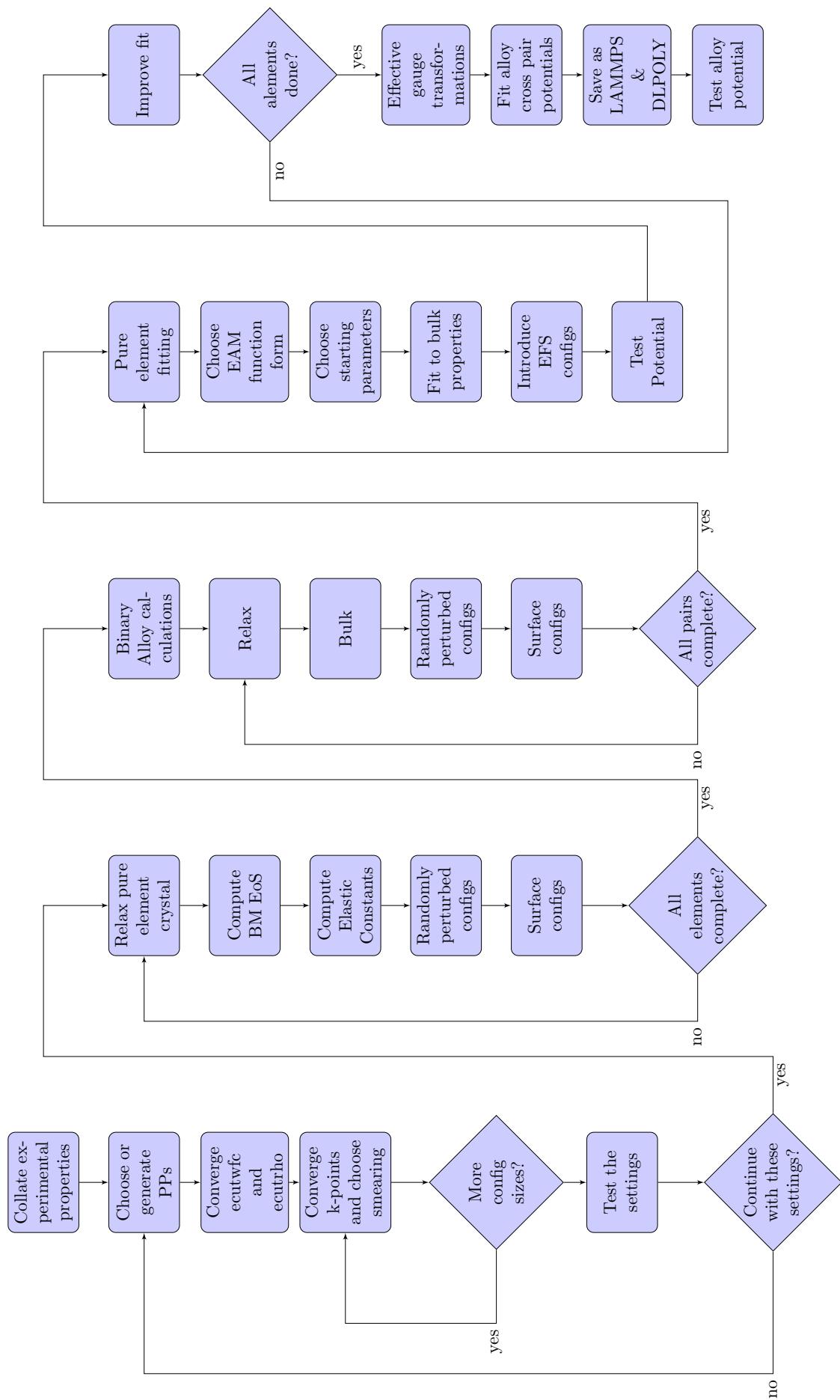


Figure 8.1: Work flow used to fit potentials

## 8.2 Experimental & DFT Bulk Property Data

The bulk properties for Fe (BCC) and Pd are available on several websites and in published work. The lattice parameter, bulk modulus and elastic constants are available for Fe BCC and Pd FCC, but as pure Fe FCC is theoretical under normal conditions, this is computed here using DFT.

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. The lattice parameter for each element, using the density of that pure element in its state under normal conditions, is predicted for that element either for both the FCC and BCC structures (table 8.1).

Element	Atomic Mass	Density kg/m <sup>3</sup>	Atoms/m <sup>3</sup>	FCC (Bohr/Angstrom)	BCC (Bohr/Angstrom)
Al [180]	26.98	2700	$6.02 \times 10^{28}$	7.66/4.05	6.08/3.22
Cr [181]	52.00	7140	$8.27 \times 10^{28}$	6.89/3.64	5.47/2.89
Fe [181]	55.84	7874	$8.47 \times 10^{28}$	6.83/3.61	5.42/2.87
Ni [182]	58.69	8908	$9.09 \times 10^{28}$	6.67/3.53	5.30/2.80
Ru [183]	101.07	12270	$7.42 \times 10^{28}$	7.14/3.78	5.67/3.00
Pd [184]	106.42	12023	$6.83 \times 10^{28}$	7.34/3.88	5.83/3.08

Table 8.1: Predicted lattice parameters based on the density, atomic number and type of structure

Aluminium appears here because it has a simpler electronic structure and no magnetic properties to complicate the DFT calculations. These calculations for Aluminium complete much faster than for the other elements listed, so it was used throughout to develop and test the computer codes created.

## 8.3 DFT Calculations

### 8.3.1 Quantum Espresso

There are a choice of several different DFT programs to choose from, including VASP, Siesta and Quantum Espresso. The open source Quantum Espresso includes the PWscf binary that solves electronic structure calculations with plane wave pseudopotentials. It calculates the total energy of a system as well as the forces between atoms and the stress on the simulation box.

As the calculations contain Iron, it is important to take into account the magnetism caused by the filling of the d-shell. PWscf gives three modes: non-magnetic, collinear spin and non-collinear spin.

### 8.3.2 Pseudopotential Selection

The pseudopotentials were downloaded from the Quantum Espresso PSLibrary with url:

<http://www.quantum-espresso.org/pseudopotentials/pslibrary>

There are several categories of pseudopotential available. The pseudopotentials may be fully relativistic or not. The DFT calculations in this work will be non-spin or collinear spin and the element with the largest number of electrons will be Palladium, with electrons in the s, p, d shells. There will be no elements used in this work with f shell electrons, so a non-relativistic pseudopotentiel will be used for each element.

There are a number of choices of exchange-correlation functional, depending on the element:

- pz: Perdew-Zunger (LDA)

- vwn: Vosko-Wilk-Nusair (LDA)
- pbe: Perdew-Burke-Ernzerhof (GGA)
- blyp: Becke-Lee-Yang-Parr (BLYP)
- pw91: Perdew-Wang 91 gradient-corrected

In the literature, LDA and BLYP are used for organic DFT calculations. LDA and GGA type pseudopotentials have been used to model solids and, in particular, the GGA type have been used to study metals as they are more reliable at calculating parameters such as  $a_0$  (see section 5.8.11). Several elements were tested to compare the results of LDA and GGA pseudopotentials to the experimental values for the lattice parameter of each.

The settings used were:

- ecutwfc: 71
- ecutrho: 430
- k-points: 9 9 9 Monhurst-Pack grid offset
- smearing: 0.04 Ry Mazari-Vanderbilt cold smearing
- calculation type: non-polarised calculation
- pseudopotentials: PZ for LDA, PBE for GGA

The results were computed as follows:

Element	Crystal	$a_0$ exp. (ang)	$a_0$ LDA (ang)	$a_0$ GGA (ang)
Al	FCC	4.05	3.98	4.04
Fe	BCC	2.87	2.69	2.75
Pd	FCC	3.89	3.84	3.93
Cu	FCC	3.61	3.49	3.59

Table 8.2: Predicted lattice parameters based on the density, atomic number and type of structure

The simpler LDA pseudopotentials consistently underestimate the lattice parameter  $a_0$  for the four metals tested, and on average they underestimate by 3%. The GGA calculations are within 1.5% of the experimental value.

### 8.3.3 Smearing Type

The material being studied here is a metal/metal alloy, and this has important consequences when setting up the calculation. The conduction and valence bands overlap in a metal, and the fermi energy passes through this overlap. When the occupied states are summed, integrating over all k-points k[185], those bands that cross the Fermi energy will drop to zero. There are two solutions used here to reduce this abrupt drop off: increase the number of k-points, or add a smearing to smooth the occupation function. Previous work has given the following suggested smearing types and values as a start point.

- Degauss values 0.01Ry Methessel-Paxton [186]
- Fermi-Dirac 0.01eV (0.00074Ry) [187]
- Marzari-Vanderbilt 0.01Ry [188]

- Degauss 0.03 + 0.05 [189]
- Marzari-Vanderbilt 0.05Ry [190]

This suggests that a range between 0.01Ry to 0.05Ry should be investigated as a sensible range of smearing energies. There are a choice of four smearing functions in QuantumEspresso.

- Gaussian
- Fermi-Dirac
- Methfessel-Paxton
- Marzari-Vanderbilt

Functions such as the Gaussian and Fermi-Dirac are not cold smearing, and will converge to the wrong energy whereas the Mazari-Vanderbilt and Methfessel-Paxton are designed to reduce and heating whilst smearing the electron wavefunction. An author of QuantumEspresso, N. Marzari, co-developed the cold smearing Marzari-Vanderbilt function, and this was selected as it had appeared a number of times in the literature and throughout the QuantumEspresso tutorials.

## 8.4 Parameter Convergence

The energy cutoff, charge density, k-point and smearing values must be selected carefully for any similar DFT calculation. There is a balancing act between the accuracy of the result of a particular calculation and computational time. This will also vary with the pseudopotential, complexity of the electron configuration of the atoms, whether or not the calculation is with or without spin, and so on.

The pseudopotentials and smearing type have already been selected, and the first set of calculations will not include magnetism. The parameters must work well with high symmetry geometries, such as a BCC or FCC crystal with the atoms in their exact position, but also where the atoms have been slightly perturbed from their exact position in the lattice.

### 8.4.1 Ecutwfc and Ecutrho

A primitive FCC or BCC is created and the atoms are slightly perturbed, breaking symmetry but also putting forces on each atom within the cell. A reasonably high number of k-points are used and the smearing value should be low, but the k-point values in particular will depend upon the computing resources you have available.

Pick a low ecutwfc and set the ecutrho equal to four times the value of the ecutwfc. Depending on the pseudopotential, the calculation may fail because the values are too small. If this is the case, continue increasing the value in small (perhaps 5Ry) increments until the calculation runs successfully.

Next, continue to increase the ecutwfc value whilst increasing the value of ecutrho such that  $ecutrho = 4 \times ecutwfc$  in steady increments (5Ry). Once the calculated force and energy per 1Ry increment fall below a threshold set by the user, the values have converged. In this work, the convergence values were:

- Energy convergence per 1Ry:  $1.0 \times 10^{-6}$  Ry ( $1.36 \times 10^{-5}$  eV)
- Force convergence per 1Ry:  $1.0 \times 10^{-5}$  Ry/Bohr ( $2.57 \times 10^{-4}$  eV/Ang)

At this point, it may be possible to reduce the ecutrho value further while remaining within this convergence threshold, so the ecutwfc value is held fixed while the ecutrho value is reduced in steps of 5Ry.

Finally, attempt to reduce the value of ecutwfc further in 1Ry steps while holding the ecutrho value steady for as long as the convergence value remains within the set threshold.

### 8.4.2 K-Points and Smearing Value

A configuration with a similar size and number of atoms as those in the calculations should be created. In this work, the reference database and bulk properties will be calculated using 32 atom configurations in an approximately sized  $7 \times 7 \times 7$  angstrom box. The Iron FCC k-points and smearing are calculated using a 32 atom FCC box with sides length 7.2 angstrom and the Palladium with a 32 atom FCC box with sides length 7.8 angstrom.

As with the ecutwfc and ecutrho convergence calculations, the atoms are slightly perturbed from their exact crystal positions, putting a force on the atoms and breaking the symmetry (as will be the case for the configurations used for fitting).

The convergence process is more complex for k-points and smearing. There might be a convergence that, at face value, looks better (a smaller convergence value and a decrease in the computation time and memory requirements) for combinations with a higher smearing value, but it is preferred to have a smaller smearing value with a higher number of k-points, while maintaining a reasonable computing time, as the greater the smearing the further the calculated value is from the actual calculated value (with no smearing and a very high number of k-points).

A 2D plot of energy and force values as a function of number of k-points and smearing may then be used to choose reasonable settings that balance the requirements.

### 8.4.3 Automating Convergence

The process requires the creation of many input files, and manually editing these files is time consuming and vulnerable to human error. Over 100 input files may be required to converge the ecutwfc and ecutrho values, with a further 50-100 required to generate the 2D k-point and smearing plots.

A python code, QECONVERGE, is developed to automate this process and plot the results automatically for the user (section 9.2).

## 8.5 Bulk Property Calculations

### 8.5.1 QEEOS Python Program

A Python code, QEEOS, (section 9.5) has been developed to calculate bulk properties of a structure using the DFT code Quantum Espresso. It is able to fit the equation of state for a cubic structure, and elastic constants for orthorhombic structures (which includes the subset of cubic structures). In brief, the user provides a template PWscf input file, and the program adjusts the crystal basis vectors, applying the homogeneous strain and 9 distortions required to calculate the equation of state and elastic crystals, as discussed earlier in this work (section 5.2.6). The output data and highlights of this code are detailed in the results (section 9.6) and appendix (??).

### 8.5.2 DFT Calculations

Although experimental data does exist for several of the structures, it was decided to calculate all the properties using DFT. This was partly to confirm that both the QEEOS and PWscf, with the converged settings, were giving reasonable values and also to give data points that would better align with the energy, force and stress configuration calculations.

Experimental data is not available for FCC Fe and FCC Ru and so it was essential to compute these. Although spin collinear equations were used throughout, it was particularly important for these settings to be used for Fe. The relaxation step allowed for a relaxation of the atom positions, the basis vector and lattice parameter. This gave cubic structures for Ru and Pd but a tetragonal structure for Fe.

The bulk property calculations were automated with the QEEOS code and this applied the preprogrammed nine strains, as discussed in section 5.2.6, and the results are discussed in chapter 9.

## 8.6 DFT Reference Database

A database of configurations generated by DFT was compiled for Iron, Palladium, Ruthenium with alloys of Iron-Palladium and Iron-Ruthenium. The potentials are derived for FCC Iron and so these configurations were also FCC.

### 8.6.1 Bulk Property Calculations

All the configurations in this work are FCC structure crystals. The QEEOS program (section 8.5.1 and 9.5) was used to compute the relaxed lattice parameter and primitive cell basis vector.

This program also automated the computation of equation of state, bulk modulus and elastic constants. A 32 atom supercell was used for each pure element where the bulk properties were computed. Ideally a large supercell or a larger number of k-points would have been used but throughout this work the number of processor cores per node ranged from 20-72 and this along with time constraints limits complexity of the calculations.

The data output from the QEEOS code forms a part of the reference database that is discussed in chapter 9.

### 8.6.2 Isolated Atom Settings

Density Functional Theory solves the Kohn-Sham equations for an infinite repeating lattice, and at first it doesn't seem applicable to single atoms or molecules that are not part of much larger lattice. There are settings that may be used to compute the energy of an isolated atom and these are discussed on the Quantum Espresso forum[191] and Materials Square[192].

The settings of note used in these calculations were a configuration of just 1 isolated atom in a large cubic cell (20+ Bohr). The computations were all nspin = 2 to ensure the electrons are computed spin up and down. The assume\_isolated setting is given the makov-payne setting, as the system is a cube.

It was important to reduce the number of processor cores as too many would cause matrix decomposition errors for a low number of atoms, causing Quantum Espresso to crash. An option to leverage more cores was to split the calculation into pools using the -npool input flag for PWscf, but this was not necessary for this set of calculations.

### 8.6.3 Pure Element Defects

The potentials are created to apply to FCC Iron, Palladium, Ruthenium and alloys of Fe-Pd and Fe-Ru. The configuration database will include several defects:

- octahedral interstitial
- tetrahedral interstitial

- 100, 110 and 111 interstitials
- crowdion
- a vacancy

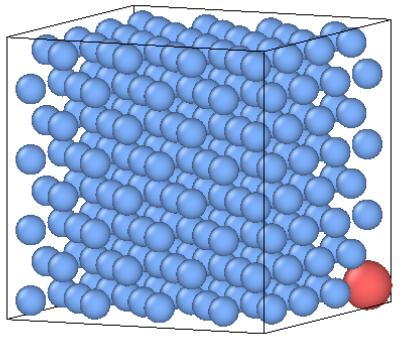
The configurations for these defects were estimated using the perfect crystal positions of 2x2x2 supercells containing 32 atoms. The defect or vacancy was applied, in some cases changing the total number of atoms, and the defect position was estimated.

A number of calculations, and the chance of a successful convergence in both the electronic state SCF calculation and relaxation calculation, were very sensitive on the input parameters. Aluminium was not as sensitive and was used to relax the starting configuration. This was then used as the starting point for either Fe, Pd or Ru.

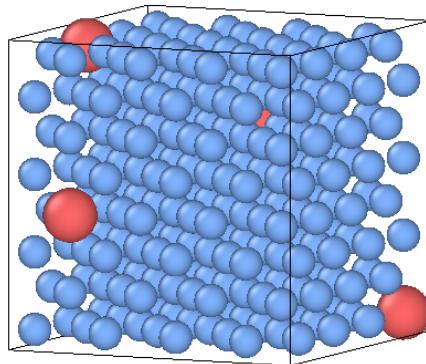
Due to the periodic nature of these calculations, having only 32 atoms was not ideal as each defect cell would be very close to another. Unfortunately, as stated a number of times, it was a constraint that was needed to be able to complete the work in a reasonable amount of time.

#### 8.6.4 Alloy Atom Positions for Pd and Ru

The stainless steel being studied is to be doped with 1% or less of a platinum group metal. Ideally a super cell with at least 256 atoms (4x4x4 primitive FCC cells) would be used for all calculations, but due to the complexity of the calculations the majority were 2x2x2 super cells containing just 32 atoms.



(a) Single atom, 256 atoms in total



(b) Multiple atoms, 256 atoms in total

Most of the configurations were prepared using the relaxed FCT structure computed for Fe. One or more iron atoms were then replaced with either Pd or Ru. The majority of calculations were carried out using the 32 atom configurations. The larger 256 atom calculations were better suited to represent the low percentage of PGM but were only computed using the gamma k-point (figure ??). This was another compromise in order to complete the calculations in time.

#### 8.6.5 Configurations with Randomised Positions

Many of the atom configurations generated up to this point are symmetric and, whilst they have a variety of energies and stresses to use for fitting, the forces on the atoms are almost balanced and are small or none. A large set of configurations with positions perturbed slightly from the exact lattice position are created.

Rather than use a linear random distribution a Gaussian distribution is used. Some atoms had a larger perturbation but the majority are be slightly closer to the exact position, as would be expected with this type of distribution. The maximum amount of perturbation was selected randomly between a minimum and maximum value.

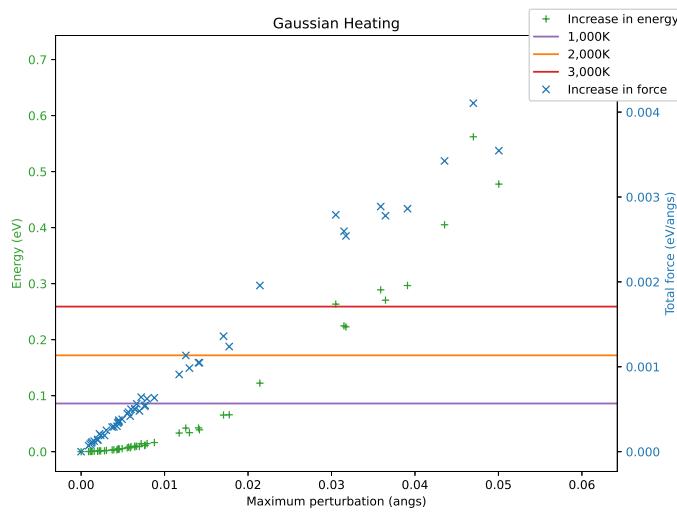


Figure 8.3: Effect of randomly perturbing atom locations

As the amount at which the atoms are perturbed increases, the energy and total force between the atoms increases. If the disturbance increases too much, it will likely be useless for our needs as it would be beyond the temperatures relevant to this work, and it may also cause the SCF calculations to fail.

In the case of the alloy configurations, the randomised positions were loaded into the files. In all cases convergence failed. As perturbed atom positions had already been computed for Fe a new set of input files were created holding the Fe atoms in their perfect locations. The added Pd and Ru atoms were then perturbed slightly from their perfect position within the lattice in order to compute forces and energies at slightly different positions.

### 8.6.6 Surface Energy Calculations

One key aspect of this work is to determine whether or not either Pd or Ru are depleted from the surface at grain boundaries under irradiation. With this in mind, the surface energies for the pure elements were calculated by creating a slab of atoms with an open surface on each side.

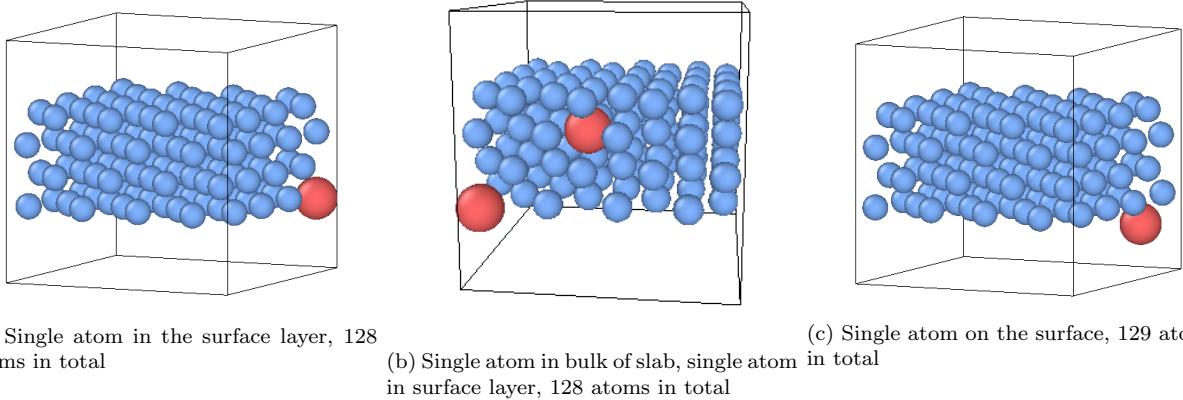


Figure 8.4: A sample of configurations used for binary alloy DFT calculations for Fe-Pd and Fe-Ru.

A set of 32 atom configurations are used where the separation between the two facing surfaces is varied. Several more 128 atom configurations are also prepared for the three pure elements. To investigate the effect of a small addition of either Ru or Pd in the alloy a slab of Fe was prepared with Ru or Pd either within the slab, at the surface or on the surface (figure 8.4). These examples were then added into the reference database used to fit

the binary potentials.

### 8.6.7 Defects in Fe-Pd and Fe-Ru Alloys

Attempts were made to compute simple defects in the alloy but the majority failed to converge, either in the electronic structure SCF iteration or a failure to relax geometrically that then lead to a SCF failure. Rather than focus too much on this, more time was spent investigating Ru and Pd in a slab of Fe where the calculations were more successful.

## 8.7 DFT Errors, Convergence Failures and Repeat Calculations

### 8.7.1 Insufficient Memory

The DFT calculations are performed on a supercomputer with many processors and many cores per processor. The program uses two implementations of parallelization.

- OpenMP - many threads have access to the same shared memory
- OpenMPI/MPICH - many processes, each with it's own memory

Whilst some favourable results have been shown using a hybrid of OpenMP and OpenMPI, it was more straightforward to use in OpenMPI mode only in this work. Unfortunately, the amount of memory per process overwhelmed the computing nodes for certain calculation. The job script was modified to reserve entire nodes and all the memory on each node, but to only use the cores on one of the two processors to halve the processes per node, but double the memory per process. It was quicker to do this than wait in the high memory queue on BlueBEAR, where the memory per process would be perhaps double than that required.

### 8.7.2 Scratch Restrictions

When the calculations first started, the scratch space was located on a large shared space on the BlueBEAR computer. Over time the drive space quota was restricted and the scratch was relocated to individual nodes. There were few issues with less complex calculations as the SCF temporary data was several GB in size with the scratch space on each node being approximately 120GB. Once a batch of collinear iron calculations had been submitted to the job queue, they would quickly fill the scratch drive and the job would terminate after just a few calculations, as the node had run out of scratch drive space.

Listing 8.1: Sbatch file

---

```

1 #!/bin/bash
2 #
3 #SBATCH --job-name=feforfit
4 #SBATCH --output=jobout.txt
5 #SBATCH --account=readmsd02
6 #
7 #SBATCH --ntasks 40
8 #SBATCH --nodes 1
9 #SBATCH --time 1800:00
10 #SBATCH --mem 180GB
11 #
12 #SBATCH --get-user-env
13 #SBATCH --export=NONE
14 #

```

```

15 unset SLURM_EXPORT_ENV
16
17 module purge; module load bluebear
18 module load bear-apps/2018a
19 module load iomkl/2018a
20 module load Python/3.6.3-iomkl-2018a
21 module load matplotlib/2.1.1-iomkl-2018a-Python-3.6.3
22
23 # Change to $PBS_O_WORKDIR
24 cd "$PBS_O_WORKDIR"
25 # Set the number of threads to 1
26 export OMP_NUM_THREADS=1
27 export PROC_COUNT=40
28 export PWSCF_SCRATCH=/scratch/bxp912
29 export PWSCF_PP=/rds/homes/b/bxp912/pp
30 export PWSCF_CACHE=/rds/homes/b/bxp912/pwscf_cache
31 export PWSCF_BIN=/rds/homes/b/bxp912/apps/qe-6.3/bin/pw.x
32 export PWSCF_SCRIPT=/rds/homes/b/bxp912/apps/qe-6.3/bin/pw.sh
33
34 python /rds/homes/b/bxp912/apps/python/qeforfit.py input.in > results.txt

```

---

Listing 8.2: Wrapper file

```

1#!/bin/bash
2rm -R $PWSCF_SCRATCH/*
3mpirun -n $1 $PWSCF_BIN -in $2 > $3

```

---

As the convergence and equation of state/elastic constant python programs automatically created and submitted the PWscf jobs, they would need to be continually restarted after clearing out the scratch. To work around this a batch script (listing 8.1) and wrapper script (listing 8.2) for pw.sh were written. The batch script sets the environment variables required for the sbatch job scheduling program and the wrapper script, and the wrapper script then clears the scratch directory before running PWscf.

### 8.7.3 Caching Repeat Calculations

Several of the programs required many hundreds of DFT calculations. At first, if an error was encountered and the program had to start from scratch, computation time would be wasted. To solve this, a directory was set to cache input and output files for PWscf. A hash would be generated from the input file and this would be used as the cache file name. If a calculation was successful, it would save the input and output file. In future, if the same calculation was submitted again, the cached file would be returned. The cache could be copied from BlueBEAR to other computers to save on repeat submission to the BlueBEAR job scheduler.

### 8.7.4 Not Straying Too Far From Reality

It is important to provide the calculation with reasonable and realistic configurations and settings. Where a pure element such as Aluminium, Iron or Platinum was being investigated, the known experimental lattice parameters were used as a starting point (table 8.1 and table 8.2). If the parameters were too far from these values, for example inputting a lattice parameter of either 2 or 8 angstroms for Aluminium rather than 4, as a starting approximation, the DFT calculation might struggle to relax the structure, or to even successfully run or converge a calculation.

The FCC Iron structure lattice parameter was estimated from the density of FCC steel, and the lattice parameter for alloys could similarly be estimated within a reasonable margin to use as a start point for the calculation.

### 8.7.5 Adjustment of DFT Parameters

Throughout the DFT segment of this work, there were a number of issues that either resulted in the calculations failing or completing and not converging. There were a number of causes for this and several solutions were used throughout the process to converge the SCF calculation.

Where the ecutwfc and ecutrho parameters are not converged there is a danger that values too low therefore containing too few plane waves in the basis set. This might cause problems with the SCF causing it not to converge. The same can be said about a failure to pick appropriate k-points and smearing.

There are a selection of mixing modes available in PWscf, giving a choice of Thomas-Fermi screening for homogeneous systems, local density dependent Thomas Fermi (local-TF) screening for inhomogeneous screening and Broyden mixing as the default.

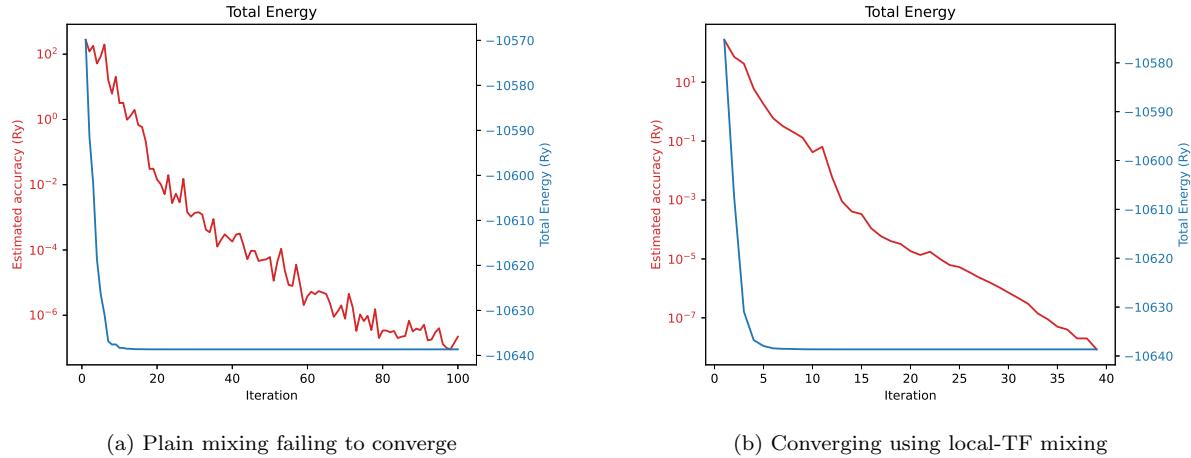


Figure 8.5: Fe-Ru calculations failed to converge in 100 iterations when plain mixing was used, but achieved convergence in under 40 iterations with local-TF convergence

For calculations with defects the local-TF setting was used and for equation of state and elastic constant calculations, where the distribution of atoms is more homogeneous, plain Broyden mixing was used. In most circumstances the mixing factor (mixing\_beta) was reduced to 0.1. This wasn't always the case, and for certain defect calculations the default value of 0.3 proved to be the more efficient choice.

One other important parameter, that primarily affects geometry and position relaxation calculations (VC-RELAX), was the convergence threshold for the SCF calculations (conv\_thr). The default value is  $1.0 \times 10^{-6}$  Ry and this was sufficient for SCF calculations and most VC-RELAX calculations. Several relaxation calculations failed to converge and, at the suggestion of authors of Quantum Espresso, the threshold was reduced to  $1.0 \times 10^{-7}$  Ry. This would lead to a longer time to compute the first geometry optimisation step. As this result affects future steps the success of convergence for subsequent steps and the time taken to complete the optimisation was reduced.

### 8.7.6 Trial Parameters for Iron with and Octahedral Defect

Calculations where there is a high level of symmetry finished relatively quickly and without much trouble. Where the position of each atom was perturbed slightly or where there was a defect the SCF calculations would take more iterations to converge and would often fail completely. Where the geometry and atom position was also being optimised through repeated sets of SCF iterations the calculation would begin to converge but would ultimately fail through SCF non convergence.

Iron with an octahedral defect was used to help fine tune parameters. Initially a set of SCF calculations were performed with different parameters, and finally these were used to relax the cell geometry and atom positions

Nbnd	Ntype	K	Degauss	Scfconv	Beta	Ndim	Mode	Iterations
317	5	9x9x9	0.04	$1.0 \times 10^{-6}$	0.1	10	Plain	75
320	3	9x9x9	0.04	$1.0 \times 10^{-6}$	0.05	10	Local-TF	46
317	3	9x9x9	0.04	$1.0 \times 10^{-7}$	0.1	10	Local-TF	42
340	3	9x9x9	0.04	$1.0 \times 10^{-6}$	0.8	8	Local-TF	38
320	3	9x9x9	0.04	$1.0 \times 10^{-6}$	0.1	10	Local-TF	36
317	3	9x9x9	0.04	$1.0 \times 10^{-6}$	0.8	8	Local-TF	33
317	5	9x9x9	0.04	$1.0 \times 10^{-6}$	0.3	8	Local-TF	32
317	3	9x9x9	0.04	$1.0 \times 10^{-6}$	0.3	8	Local-TF	32
320	3	9x9x9	0.04	$1.0 \times 10^{-6}$	0.3	10	Local-TF	29
320	3	11x11x11	0.04	$1.0 \times 10^{-6}$	0.3	10	Local-TF	29
320	3	13x13x13	0.04	$1.0 \times 10^{-6}$	0.3	10	Local-TF	29
320	3	9x9x9	0.07	$1.0 \times 10^{-6}$	0.3	10	Local-TF	25

Table 8.3: Improving convergence SCF calculations of defects in Iron

for this defect.

Changing the number of k-points did not have an impact on the number of iterations required to converge, but changing the smearing value did. An increase in smearing would be detrimental to the accuracy of convergence to the correct value and by doing this it would cause a difference in energy between past calculations with a smearing of 0.04Ry and those with 0.07Ry.

The adjustment that had the largest impact on reducing the number of iterations was the change from plain mixing to Local-TF. Following this, using a mixing beta of 0.3 where more weight is given to the previous state, also helped to reduce the number of iterations.

The attempts to optimise the values was not exhaustive as there were constrictions on time. However, despite making these improvements and reducing the convergence threshold to  $1.0 \times 10^{-8}$ , the cell geometry and position relaxation of the starting configuration for an octahedral defect (by adding an atom at the octahedral location) also fails to relax by SCF non-convergence within an iteration of the relaxation algorithm.

The optimization of the geometry and atom positions was likely due to it being a complex system to model with the initial configuration too far from the relaxed one. To work around this the octahedral defect for a more simple metal, Aluminium, was relaxed to give a better starting point for the Iron calculation. The increase in cell size was also applied. With a better initial configuration and the adjusted parameters, the Iron octahedral calculation successfully completed.

This process was repeated for tetrahedral, 100, 110, 111 and crowdion defects.

## 8.8 EAMPA: Potential Analysis and Fitting Code

A python-fortran based computer code was developed to fit interatomic potentials to bulk properties and DFT calculated forces and energies. It has been designed to take advantage of both Fortran and Python. Fortran is used to compute the neighbour lists, energies, stresses and forces, and Python is used to read and write data, produce plots and control the potential fitting. This allows the use of custom optimization subroutines as well as the use of those included in the SciPy module. Highlights from the code are discussed in the results (section ??) and the appendix of this work (appendix ??).

## 8.9 Investigating the Transferability of Existing Potentials

There are currently a number of EAM Iron potentials, as well as a MEAM potential for Fe-Pd. Several pure element potentials also exist for Ruthenium and Palladium. Whilst it would be convenient to have a one size fits all potential, this is not the case. Taking the Ruthenium potential as an example, in work that required the reproduction of stacking fault and surface energy of the element, the existing potentials were inadequate and a new potential was developed[193].

Several potentials were selected to test transferability. First, the FCC Sheng potential for Aluminium was applied to both FCC and BCC structures in the EAMPA code. The properties of FCC Aluminium are known, and those for BCC Aluminium were calculated using DFT. This investigation shows that some properties, such as the lattice constant, does transfer quite well to the new structure, but the bulk modulus and  $C_{11}$  elastic constant are both poorly reproduced (table 8.4).

Property	Al FCC Exp.	Al FCC Sheng	Al BCC DFT	Al BCC Sheng
a0	4.05	4.02	3.23	3.18
e0	-3.36	-3.36	-3.26	-3.27
b0	76	86	69	50
$C_{11}$	114	124 (125)	47 (47)	11 (5)
$C_{12}$	62	67 (66)	81 (24)	70 (63)
$C_{44}$	32	29 (32)	40 (12)	29 (42)

Table 8.4: Applying the Sheng Al potential, developed for FCC crystals, to the BCC structure. Elastic constants are calculated using the method by Mehl et al with values from the method by Ravindran et al in brackets.

The Mendelev 2003 potential was developed to fit the properties of BCC Iron, but also to reproduce the transition from BCC to FCC at high temperatures. However, when using this potential for low temperature FCC Iron, it does not reproduce the properties predicted by DFT. All 9 orthorhombic elastic constants have been omitted in table 8.5, but there is a significant difference for the three elastic constants given as well as the bulk modulus.

Property	Fe BCC Exp.	Fe BCC Mendelev	Fe FCC DFT	Fe FCC Mendelev
a0	2.86	2.86	3.42	3.59
e0	-4.32	-4.13	-4.26	-4.01
b0	170	176	226	98
$C_{11}$	243	255 (260)	364	6 (79)
$C_{12}$	145	137 (143)	142	145 (69)
$C_{44}$	116	114 (111)	186	86 (48)

Table 8.5: Applying the Sheng Al potential, developed for FCC crystals, to the BCC structure. Elastic constants are calculated using the method by Mehl et al with values from the method by Ravindran et al in brackets.

Finally the Ackland 1997 potential for Iron was tested. This older potential gives a better value for the cohesive energy when transferred to the FCC structure, but other properties such as the relaxed lattice constant are quite different (table 8.6).

This investigation into existing potentials shows that whilst some of the properties are replicated, many do not match the DFT computed values. This supports the derivation of new potentials to be fit that replicate the DFT values and take into account the characteristics of a number of provided configurations. Additional plots are included in the appendix that show the expected equations of state compared to those generated using the above potentials (section ??).

Property	Fe BCC Exp.	Fe BCC Mendelev	Fe FCC DFT	Fe FCC Mendelev
a0	2.86	2.87	3.42	3.62
e0	-4.32	-4.32	-4.26	-4.26
b0	170	200	226	141
$C_{11}$	243	265 (248)	364	117 (140)
$C_{12}$	145	168 (151)	142	153 (117)
$C_{44}$	116	125 (128)	186	164 (94)

Table 8.6: Applying the Sheng Al potential, developed for FCC crystals, to the BCC structure. Elastic constants are calculated using the method by Mehl et al with values from the method by Ravindran et al in brackets.

## 8.10 Fitting Potentials with EAMPA

The custom computer code EAMPA was developed in order to fit the parameters for the potentials to the DFT reference database. It was a long process that utilised nodes on the University of Birmingham BlueBEAR computer. At the time of writing, the code was able to fit energy, forces and bulk properties. I was not convinced that the computation of stress in EAMPA and in PWscf were identical and so throughout I set the weighting of stress to zero.

### 8.10.1 Choice of Function and Functional

A number of functions and functionals have been used previously and these are discussed in detail in chapter 5. Several were used to attempt to fit the potentials, but eventually the cubic splines and tight binding type functional, as used by Mendelev, Ackland, Bonny et al, were selected.

Initially, I looked to the computer program DIRAC for the density function, as it computes the radial electron density distribution (figure 8.6 and figure 8.6). Unfortunately, as may be expected, there were a number of issues in using the density distribution of a single atom. The atomic separation for both iron and palladium, for a relaxed crystal, begins at approximately 2-2.5 angstrom, but most of the electrons, from all shells, are distributed much closer. The plot does not take into consideration the valence electrons shared in the crystal, bonding the metal atoms together.

From the point of view of the potential, the pair potential will dominate at small separations due to the ZBL function. The density function will be important at separations of 2 angstrom up to the cutoff radius (5-6 angstrom). For this reason, the idea to use the DIRAC computed densities was abandoned in favour of a cubic spline.

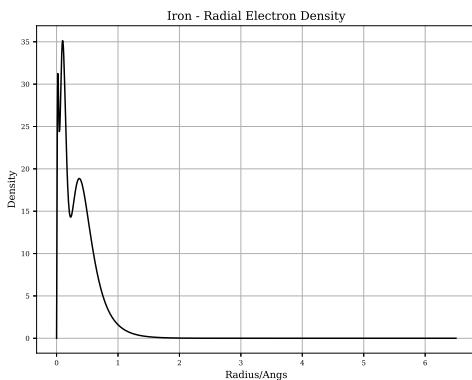


Figure 8.6: DIRAC calculated radial electron density for Iron

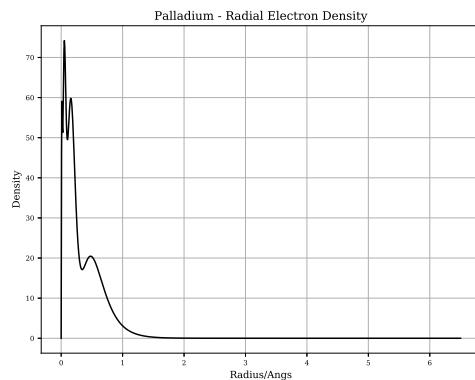


Figure 8.7: DIRAC calculated radial electron density for Palladium

One interesting feature that Bonny used in his code was to add a hard core spline to the pair potential with a positive coefficient. This was to ensure that at small separations, say below 1.5-1.8 angstrom, the atoms would repel. The cubic spline used in this work for the density also has a fixed positive coefficient to help stop negative densities for small separations. This is dependent on the number of cubic splines used, and is a feature that can be turned off in the EAMPA code.

A cubic spline was also used for the pair potential, and as with Bonny's code, this had a hard core potential but in addition to this a ZBL core was added to the entire potential to ensure the exponential increase in energy as modelled with the ZBL function. This was to ensure the potential would both model the material when out of a radiation field, where the atoms are several angstrom apart, but also able to capture a damage cascade where the separation drops and the ZBL becomes the dominant feature, over both the pair cubic spline and the embedding functional.

There was a choice of several embedding functionals but one based on the Finnis-Sinclair and EAM was used. It consisted of the square root term but also a zeroth power and square term for greater flexibility. The addition of a fourth power term had been trailed, but occasionally when fitting the parameters the coefficient of the fourth power would become negative leading to a negative energy at high electron densities.

## 8.10.2 Searching Parameter Space

Improving the potential was an iterative process that required multiple runs of EAMPA feeding the last potential as a starting point for the next potential. In the first instance, the exact bulk property fitting code of Bonny was used, but it did not always find the parameters that reproduced the data points it was given. This was perhaps due to a lack of data and knowledge on how to use the code effectively, but it did give a starting point with a modest number of cubic splines for the density and pair functions.

These initial set of parameters were then used with EAMPA. The first task with this code was to find parameters that replicated the energies, forces and bulk properties for each of the pure elements.

The type of search could be adjusted for each fitting job, but the typical process was as follows. A short random search through parameter space followed by a small simulated annealing search with a wide search area. This was then followed by a genetic algorithm search with quite a wide search area. Next a second genetic algorithm search over a smaller range of parameters but with more iterations and a larger population, and this step used most of the computation time. Finally a short simulated annealing and local BFGS minimizer were used before outputting optimised parameters. A sample input is given in listing 8.3, but the user is free to add fewer or more steps and search types.

Listing 8.3: Parameter search settings when fitting in EAMPA

---

```

1 #FIT type=RAND niter=1000
2 #FIT type=SA niter=1000 titer=10 tstart=100.0 tend=0.001 pfact=0.5 pvar=10.0 vartype=add gaussian=True
3 #FIT type=GA niter=200 popsize=500 fresh=0.2 search=w
4 #FIT type=GA niter=5000 popsize=2000 fresh=0.2 search=n
5 #FIT type=SA niter=1000 titer=15 tstart=1.0 tend=0.001 pfact=0.5 pvar=0.01 vartype=add gaussian=True
6 #FIT type=BFGS

```

---

The code has an option to set the random seed such that every job with that seed and settings will return the same output. This allowed ten or more jobs to be run in parallel, with 4 cores per job, and a different seed for each job. Typical run times were 3-5 days per iteration and several iterations required.

At the start of the fitting process all available configurations were used, but it turned out to be difficult for the code to satisfy all the constraints given the simplicity of the EAM potentials. Towards the end of the process many of the configurations were removed with bulk and surface/slab configurations being left. This was in order to train the potential specifically to these configurations in favour of the entire set.

Once the pure potentials for Fe, Pd and Ru were determined, they each underwent an effective gauge transform, with the transform being that used by Bonny et al[114][122]. Due to this transform, they were converted from analytic potentials to numeric.

The fixed numeric functions were then used in the binary alloy fitting where the only parameters adjusted are those for the cross pair potential. The searching method was similar to that used for the pure elements, but the configurations were changed to those containing both species of atom. The bulk property fitting was also removed as this fitting was only applicable to the pure elements.

## 8.11 DL\_POLY Contribution

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.

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). The code was altered with the addition of an array used to store the d-band density function and d-band embedding functional, with the s-band function/functional using the original density function and embedding functional arrays. The calculation of energy and forces on atoms was also altered to make use of these arrays where a two band embedded-atom method potential is used.

Listing 8.4: DL\_POLY 4.05 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

```

---

Extracts of the modified DL\_POLY code are included in appendix ??.

# Chapter 9

## Results: DFT Reference Database

*The key parameters required for Quantum Espresso are converged and selected in preparation for the upcoming calculations. A set of atom configurations for FCC crystals are created and DFT is used to computer the energy, forces and stresses on these crystals.*

*FCC iron does not exist at normal room temperature and pressure, without the addition of a austenite stabiliser. A set of DFT calculations are used to compute the bulk modulus, lattice parameter and cohesive energy of pure Fe FCC. Ruthenium has a HCP structure and similarly DFT calculations are used to compute bulk properties of FCC Ruthenium.*

*The database of bulk properties and atom configuration are used in chapter ?? to derive potentials for both Fe-Pd and Fe-Ru as well as for the pure elements.*

### 9.1 Introduction

While computer technology has advanced almost in accordance with Moore's law, the calculations involved to approximately simulate simple structures with Quantum Mechanics are barely accessible. Ideally, crystal structures with several hundred atoms, containing Fe, Cr, and Ni with Ru or Pd would have been used to derive a potential describing all combinations of elements.

The computer used did not have the resources required for such calculations, and so the model was simplified to either Fe-Pd or Fe-Ru. The ferromagnetism and antiferromagnetism of Fe and Cr were explored using collinear spin DFT calculations. Due to the improvement in modelling given by including magnetism, all calculations for the reference database were collinear spin.

A number of computer codes were developed to automate processes. QECONVERGE creates input files and calls PWscf to converge parameters used in this work (section 9.2). QEEOS also creates input files for PWscf but computes bulk properties such as the equation of state, bulk modulus and elastic constants (section(9.5).

### 9.2 QECONVERGE Python Code

#### 9.2.1 Purpose of Code

The process of converging cutoff parameters is one that may be automated. A program was developed 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 detail 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 provided by the user. 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 fixed 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 collection 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, if desired.

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. If the program runs multiple times, the cached output files are used rather than recalculating every input file.

### 9.2.2 Source Code and Instructions

The source code and instructions on how to use the program are available to download from GitHub.

<https://github.com/BenPalmer1983/qeconverge>

## 9.3 Convergence of Key DFT Settings and Testing

### 9.3.1 Pseudopotential

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 then used to converge the ecutwfc, ecutrho, k-points and degauss smearing values for each of the pseudopotentials required.

As discussed in previously (section 5.8.11) the GGA type pseudo-potential more reliably calculated the bulk properties of a material when compared to LDA/LSDA, and a copy of the earlier table is given in table 9.1.

Pseudo-potential	Element	a/angs	$B_0/\text{GPa}$
Experimental	Na	4.29[157]	6.3[157]
Na.pz-spn-kjpaw_psl.1.0.0	Na	4.06	8.72
Na.pbe-spn-kjpaw_psl.1.0.0	Na	4.20	7.67
Na.pbesol-spn-kjpaw_psl.1.0.0	Na	4.17	7.50
Experimental	Al	4.05[158]	76[158]
Al.pz-nl-kjpaw_psl.1.0.0	Al	3.98	78.6
Al.pz-n-kjpaw_psl.1.0.0	Al	3.98	78.6
Al.pbe-nl-kjpaw_psl.1.0.0	Al	4.04	91.3
Al.pbe-n-kjpaw_psl.1.0.0	Al	4.04	75.0
Al.pbesol-nl-kjpaw_psl.1.0.0	Al	4.01	87.7
Al.pbesol-n-kjpaw_psl.1.0.0	Al	4.01	79.0

Table 9.1: Experimental vs LSDA vs GGA - the DFT values were computed with a PWscf[159] using a 2x2x2 cell, 7x7x7 kpoints and ecutwfc=50

The following pseudo-potential files were used for each element:

- Aluminium: a plane wave GGA potential Pd.pbe-spn-kjpaw\_psl.1.0.0.UPF
- Chromium: a plane wave GGA potential Cr.pbe-spn-kjpaw\_psl.1.0.0.UPF
- Iron: a plane wave GGA potential Fe.pbe-spn-kjpaw\_psl.1.0.0.UPF
- Palladium: a plane wave GGA potential Pd.pbe-spn-kjpaw\_psl.1.0.0.UPF
- Ruthenium: a plane wave GGA potential Ru.pbe-spn-kjpaw\_psl.1.0.0.UPF

For each pseudo-potential, the ecutwfc, ecutrho and kpoint and degauss settings were converged. Ideally, the k-points settings would have been increased, but doing so would have been prohibitive due to the amount of time taken to perform each calculation.

By the time calculations were being executed for a 32 atom supercell, one node of 20 processor cores would run for several days up to a week or more, requiring the calculation to be submitted to the job queue multiple times. Al is also considered in the results section, as it is a simpler atom to run trial calculations with, and to compare to experimental results.

### 9.3.2 Collinear Spin Calculations

The relaxed crystal lattice parameters were calculated for Fe, Cr, Pd and Ru. Cr will not be used in either the Fe-Pd or Fe-Ru fitting calculations. However, there is a large increase in computation time between non-spin and collinear spin calculations. The ferromagnetic structure of BCC Fe, antiferromagnetic structure of FCC Fe and BCC Cr were investigated.

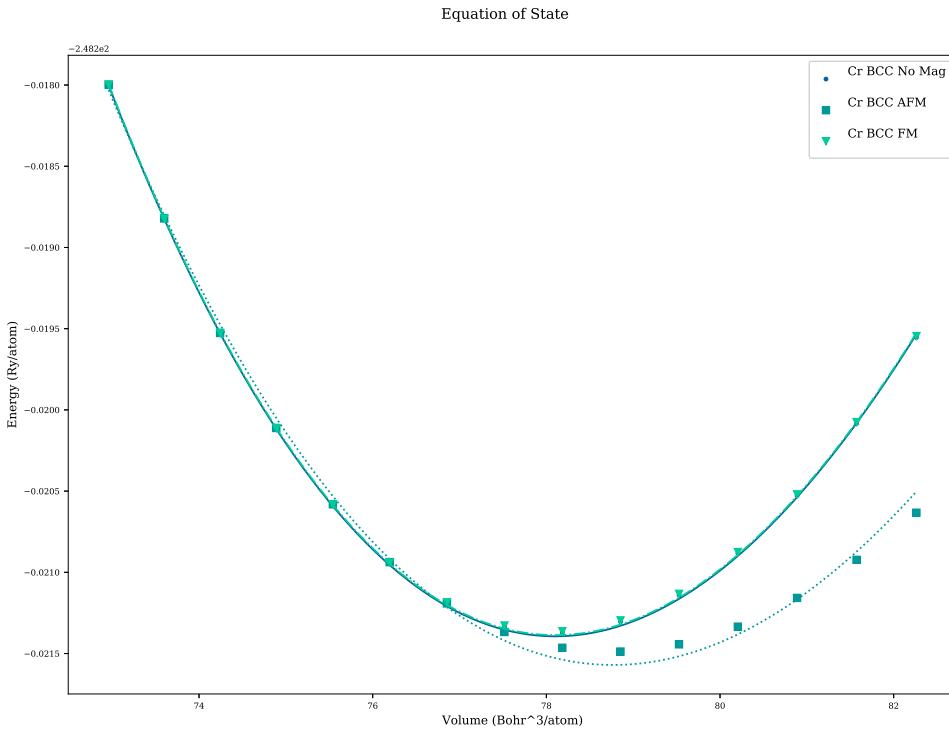


Figure 9.1: Equation of state fit through data points for Chromium BCC with no magnetism, ferromagnetic and anti-ferromagnetic configurations

Description	$A_0$ (ang)	$V_0$ (bohr3)	$B_0$ (GPA)	$E_0$ (Ry) (DFT Only)
Exp.	2.91	83.23	160	-
AFM	2.86	78.77	217	-248.2216
FM	2.85	78.10	267	-248.2214
No Mag	2.85	78.102	267	-248.2214

Table 9.2: Chromium properties with and without collinear spin

The equation of state was calculated for Chromium BCC in three ways: (1) with magnetism switched off, (2) atoms set in a ferromagnetic configuration (collinear, all in the same direction), (3) atoms set in an antiferromagnetic configuration (collinear, atoms in the same cell with spin in opposing directions).

The DFT calculated values for the bulk modulus were larger than expected, but the antiferromagnetic calculation was much closer to the experimental value (table 9.2). The  $E_0$  values do not reflect the actual energies, but show the relative calculated values. The antiferromagnetic configuration gives the lowest, optimum, energy (fig. 9.1).

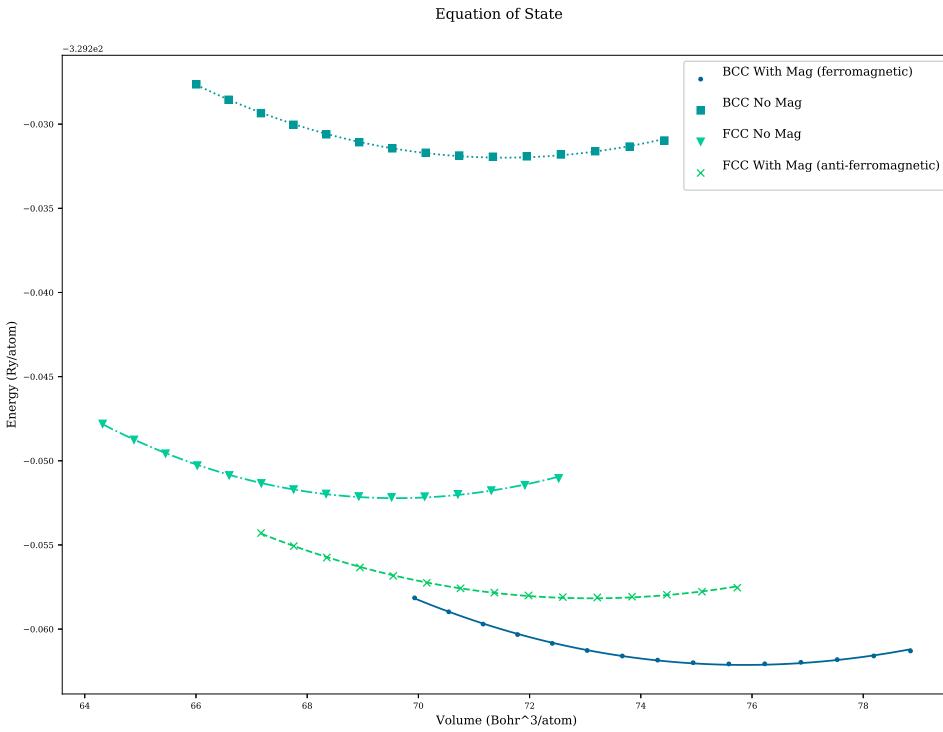


Figure 9.2: Equation of state fit through data points for Iron BCC (no magnetism, ferromagnetic) and Iron FCC (no magnetism and antiferromagnetic)

Description	A0 (ang)	V0 (bohr <sup>3</sup> )	B0 (GPA)	E0 (Ry) (DFT Only)
Exp.	2.86	79.01	170	-
No Mag	2.77	71.53	286	-329.23
FM	2.82	75.9	239	-329.26
AFM	(failed)	(failed)	(failed)	(failed)

Table 9.3: BCC Iron properties with and without collinear spin

In the case of Iron, both BCC and FCC configurations were used. The antiferromagnetic configurations for both BCC and FCC failed in that the SCF calculations failed to converge. As expected, the optimal configuration was BCC with ferromagnetism, and second to this was FCC with ferromagnetism.

Despite the increase in computational time, and other demands on resources such as scratch space and RAM per node, it was clear that collinear spin calculations were required for iron. The lattice parameter value was within 2% of the experimental value with the BCC structure set in a ferromagnetic state. Whilst the bulk modulus is only within 41% of the experimental value, this is an improvement over the non magnetic calculation that was almost 70% away from the experimental bulk modulus (table 9.3, figure ??).

### 9.3.3 Convergence Results

The QECONVERGE program was used to suggest the parameters to use, given the convergence threshold of  $1.0^{-5} RY/Bohr$  for forces and  $1.0^{-6} RY$  for energy.

Element	Pseudopotential	Ecutwfc	Ecutrho	K-points	Degauss	Atoms
Al	Al PBE KJPAW	50	200	11	0.04	32
Fe/Pd/Ru	PBE KJPAW	71	431	9	0.04	32
Fe/Pd/Ru	PBE KJPAW	71	431	Gamma	0.04	128-256

Table 9.4: DFT Settings - pseudo-potentials, ecutwfc, ecutrho, k-points, smearing

These settings in table 9.4 were used throughout the remainder of the work. The maximum values were selected from Fe, Ru and Pd as they were to be used for calculations of the pure elements and for binary allows of Fe-Ru and Fe-Pd.

The other settings used in PWscf throughout are listed in table 9.5. Several of these values were arrived at by trial and error. It was recommended by the authors of Quantum Espresso to adjust parameters such as the mixing beta if a calculation fails to converge. The mixing mode was also changed from time to time, but overall the plain mode seemed to work best in most cases.

Parameter	Value
etot_conv_thr	0.0001
forc_conv_thr	0.001
conv_thr	$1.0 \times 10^{-8}$ to $1.0 \times 10^{-6}$
diagonalization	david
mixing_beta	0.1 to 0.3
mixing_mode	plain (local-TF for defects or randomised configurations)

Table 9.5: DFT Settings - other settings

The choices of parameters are supported by the convergence plots produced by the QECONVERGE code in figures ?? and ?? (appendix ??).

## 9.4 Preliminary Calculations

### 9.4.1 Relaxed Crystal Calculations

The cohesive energy is an important value in relation to the interatomic potential. The energies calculated by DFT code depend on many factors including the energy under which plane-wave are cut-off, the degauss value, k-point settings and the SCF convergence parameters. What is more important in the DFT calculations is the difference in energy between calculations. As the cohesive energy is known, a DFT calculation may be performed for each species of atom to determine the relaxed energy. This value may then be used to calibrate other DFT calculations, so they are given relative to the energy of each atom spaced infinitely far from one another, with a binding energy of 0eV. The relaxed energies, the settings used to calculate them and the adjustment to convert them to the “real” energies are given in table 9.6.

Measurement	Al FCC	Fe BCC	Fe FCC	Pd FCC	Ru HCP
Element	Aluminum	Iron	Palladium	Ruthenium	Ruthenium
Structure	FCC	FCC	FCC	HCP	FCC
Ecutwfc (Ry)	50	71	71	71	71
Ecutrho	200	430	430	430	430
K-points	11 11 11	9 9 9	9 9 9	9 9 9	9 9 9
Smearing (Ry)	0.04	0.04	0.04	0.04	0.04
NSpin	0	2	2	0	0
Starting Magnetism	None	FM	AFM	None	None
No. Atoms	32	16	32	32	32
Energy (Ry)	-1264.08749654	-5268.18846365	-10536.25040753	-16384.95803030	-6906.78722345
Energy/Atom (Ry)	-39.502734267	-329.261778978	-329.257825235	-512.029938447	-13813.30088106
Energy/Atom (eV)	-537.462273214	-4479.836349416	-4479.782555982	-6966.524743211	-431.674201466
Known Cohesive Energy (eV)	-3.36	-4.316	-4.26 (Calculated)	-3.91	-6.74
Adjustment/atom (eV)	534.102275214	4475.520349416	4475.520349416	6962.614743211	-6.62 (Calculated)
Relaxed $a_0$ (Bohr)	15.265	10.594	12.937	14.845	5866.47338
Relaxed $c$ (Bohr)	None	None	None	1.5773	None
$U_{xx}$	1.000	1.000	1.000	1.000	1.000
$U_{yy}$	1.000	1.000	1.054	1.000	0.8660
$U_{zz}$	1.000	1.000	1.000	1.000	1.000
$U_{yx}$	0.000	0.000	0.000	0.000	0.000

Table 9.6: Relaxed energies calculated in Quantum Espresso

### 9.4.2 Relaxed FCC Iron

The configuration for Fe has been referred to throughout this work as FCC, but the relaxed collinear spin-polarized DFT calculation predicts that the structure is slightly face centered tetragonal (FCT). The optimal magnetic setting is antiferromagnetic (fig. ??) and, with the spin aligned in the z-direction, the y-axis of the unit cell is approximately 5% larger than the x and z axis.

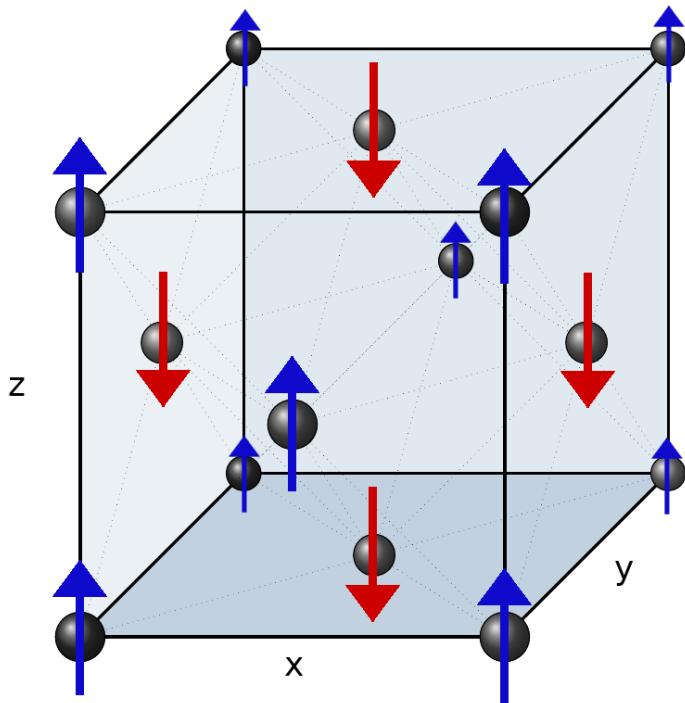


Figure 9.3: Magnetic alignment along the z-axis, antiferromagnetic Fe fcc

The atoms in the (0,1,0) plane are majority spin-up electrons, along the z-axis, and those in the (0,2,0) plane are spin-down, along the z-axis (fig. ??). Following the relaxation calculation, all the atoms were in their standard FCC positions, and the magnetic moment for each atom was 1.4901 Bohr magneton and -1.4901 Bohr magneton for the spin up and down respectively.

## 9.5 QEEOS Python Code

### 9.5.1 Purpose of Code

A code was developed to automate the process of calculating the equation of state and elastic constants of a material using Quantum Espresso. It requires an input file and a starting PWscf input file.

- the input configuration is relaxed using the vc-relax option in PWscf
- configuration files are created to compute the equation of state, and PWscf is used to calculate the energies of these configurations
- to compute the elastic constants, nine distortions are applied to the relaxed configuration with the required number of steps for each strain applied to each distortion, and these are processed with PWscf

- once all DFT work has completed, the Birch-Murnaghan equation of state is fit to the first set of energies, and the nine orthorhombic elastic constants are fit to the results of the second set of energies

### 9.5.2 Source Code and Instructions

The source code and instructions on how to use the program are available to download from GitHub.

[https://github.com/BenPalmer1983/qe\\_eos](https://github.com/BenPalmer1983/qe_eos)

## 9.6 Calculated Elastic Properties

The planewave cutoff and k-points were converged to within the required parameters for energy and force but it is known that, in general, LDA and GGA pseudopotentials under and over estimate the lattice parameters. The bulk modulus and elastic constants were calculated for FCC Aluminium and BCC Iron in order to compare to experimental values.

A summary of the main data points is given here for FCC Al, BCC Fe, FCC Pd and FCC Fe, and the first three listed are compared to experimental data. A full details of the calculated properties including plots are included in appendix ??.

### 9.6.1 FCC Aluminium

The parameters used for the Al calculations were arrived at earlier using the convergence code. The remainder of the parameters are either default settings, or were selected following trial and error (for example, reducing the mixing beta to help achieve convergence). These settings are given in table ??.

The results in table ?? show a good agreement between the experimental values for Al and the calculated values. The lattice parameter is a particularly good fit, being within 1% of the experimental value. The shear modulus does not match as well, but is within 20% of the experimental value. The RSS measurement of fit between the values was  $2.54 \times 10^2$ .

### 9.6.2 BCC Iron

The equation of state and elastic constants were calculated for BCC Fe, with no-spin and collinear spin. The settings used for the calculations are given in table ??.

The non-magnetic Iron calculation was particularly poor. Although the lattice parameter was within 5% of the experimental value, the structure was unstable and had negative values for both the Young's modulus and shear modulus (table ??). The RSS measurement of fit between the values for the was  $2.01 \times 10^5$ .

With collinear spin enabled, the structure is stable. The lattice parameter is still slightly under the experimental, but is now within almost 2% of the value. The bulk modulus predicted by fitting the Birch-Murnaghan equation of state is more than 40% the experimental value, but the Reuss and Voigt values derived from the calculated elastic constants are just 20% difference to the experimental value. The latter two values are calculated using the elastic constants, rather than the equation of state. This highlights the importance of using collinear spin calculations, despite the cost in computing time. The RSS measurement of fit between the values was  $1.09 \times 10^4$  giving an improvement of at least one magnitude.

### 9.6.3 FCC Iron

The gamma phase of pure iron does not exist for experimental measurements to be made, and the results here will be used to fit the FCC iron potential in chapter ???. The settings used for the calculations are given in table ??.

The structure is stable under the orthorhombic stability conditions given in section 5.2.7. Whilst there are no experimental measurements to check against, the values are sane when compared to the other calculations performed here (table ??). The  $C_{44}$ ,  $C_{55}$  and  $C_{66}$  values are slightly large when compared with those for BCC, but further DFT calculations would need to be performed to investigate this.

Property	Value used in potential fitting
Element	Fe
Structure	Face Centered Cubic
$a_0$	3.59 Angstrom
Nearest Neighbour	1.85 Angstrom
Basis vectors	$\begin{bmatrix} 0.96 & 0.0 & 0.0 \\ 0.0 & 1.00 & 0.0 \\ 0.0 & 0.0 & 0.96 \end{bmatrix}$
$E_{coh}$	-4.32 eV
$B_0$ (GPA)	226.1
$E$ (GPA)	356.8
$G$ (GPA)	144.8
Poisson Ratio $\eta$	0.23
Elastic Constants	$\begin{bmatrix} 364.6 & 141.6 & 233.8 & 0 & 0 & 0 \\ 141.6 & 298.7 & 130.4 & 0 & 0 & 0 \\ 233.8 & 130.4 & 364.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 186.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 266.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 186.3 \end{bmatrix}$

Table 9.7: Fe input parameters for fitting

### 9.6.4 FCC Palladium

The equation of state and elastic constants were calculated for FCC Pd with no-spin. The settings used for the calculations are given in table ??.

There is a good agreement between the computed and experimental values for Pd, without the need to move from non-spin to collinear spin calculations (table ??). However, in the alloy calculations the presence of Fe will dictate the necessity for collinear spin. The RSS measurement of fit between the values was  $4.12 \times 10^3$ .

Property	Value used in potential fitting
Element	PD
Structure	Face Centered Cubic
$a_0$	3.925 Angstrom [184]
Nearest Neighbour	Angstrom [184]
Basis vectors	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$
$E_{coh}$	3.91 eV [115]
$B_0$	184.4 GPA [115]
Elastic Constants	$\begin{bmatrix} 218.5 & 151.4 & 151.4 & 0 & 0 & 0 \\ 151.4 & 218.5 & 151.4 & 0 & 0 & 0 \\ 151.4 & 151.4 & 218.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 80.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 80.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80.3 \end{bmatrix}$

Table 9.8: Pd input parameters for fitting

### 9.6.5 FCC Ruthenium

Ruthenium at standard conditions exists as hexagonal close packed, but as this work is focused on FCC steel doped with PGMs, the properties are computed for FCC Ruthenium. The settings used for the calculations are given in table ??.

Property	Value used in potential fitting
Element	RU
Structure	Face Centered Cubic
$a_0$	3.809 Angstrom [184]
Nearest Neighbour	Angstrom [184]
Basis vectors	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$
$E_{coh}$	6.624 eV [115]
$B_0$	307.8 GPA [115]
Elastic Constants	$\begin{bmatrix} 471.5 & 219.1 & 219.1 & 0 & 0 & 0 \\ 219.1 & 471.5 & 219.1 & 0 & 0 & 0 \\ 219.1 & 219.1 & 471.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 245.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 245.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 245.0 \end{bmatrix}$

Table 9.9: Pd input parameters for fitting

## 9.7 DFT Configuration Database

### 9.7.1 Calibrating the Cohesive Energies

The isolated and relaxed energy values were computed and from these values the cohesive energies were calculated (figure ?? and table ??). In doing so, it became apparent that there were errors of up to 12% between the known values and those computed with DFT.

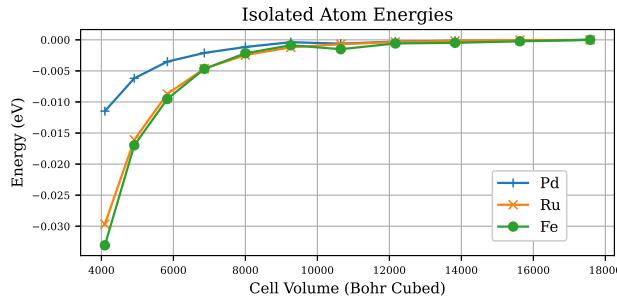


Figure 9.4: Isolated atom energies as cell size increase

An offset was needed to convert from the DFT energies to “real” energies that could be used to derive a potential.

Element	Structure	Isolated Energy (Ry)	Relaxed Energy (Ry)	Cohesive Energy (eV)
Fe	BCC	-4474.99115	-4479.83524	4.84 (exp. 4.32)
Fe	FCC	-4474.99115	-4479.78095	4.79
Pd	FCC	-6962.47261	-6966.52234	4.05 (exp. 3.91)
Ru	HCP	-5866.39894	-5873.22666	6.83 (exp. 6.74)
Ru	FCC	-5866.39894	-5873.11029	6.71

Table 9.10: DFT calculated energies per atom

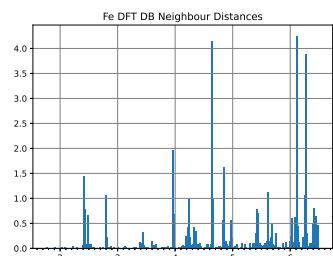
Two offsets were computed and these depend upon the settings used in the calculation (table ??). Specifically, they depend on whether 9x9x9 k-points are used, in the case of the 32 atom calculations, or the gamma point is used (128, 129 and 256 atoms).

Element	Reference Structure	Energy (Ry) (9x9x9)	Energy (Ry) (Gamma)	Cohesive (eV)
Fe	BCC	-329.2618197	-84290.00709480	-4.316
Pd	FCC	-512.0299524	-131079.09207327	-3.91
Ru	HCP	-431.6742014	-110507.45936537	-6.74

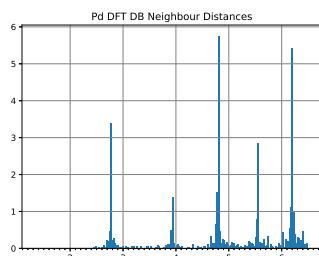
Table 9.11: Relaxed energies for 32 atoms, using 9x9x9 k-points, and 256 atoms, using the gamma point. These were used with the known cohesive energy to compute the offset energy.

### 9.7.2 Complete DFT Reference Database

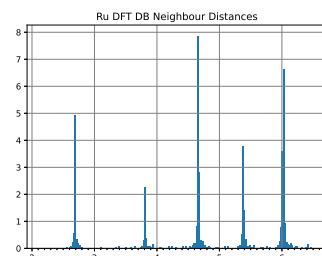
The complete database of configurations was compiled from many of the already mentioned calculations. The QEEOS code that computed the bulk properties creates PWscf input files and processes the output files. These output files are saved but the program and are as valid to use in the reference database as any. In total the database has over 600 configurations for Fe-Pd and almost 600 for Fe-Ru.



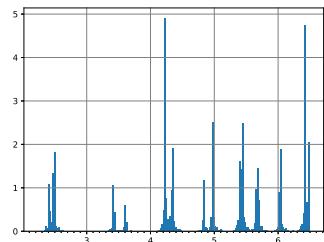
(a) Single atom, 256 atoms in total



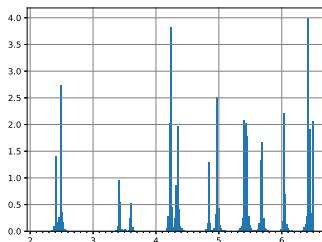
(b) Multiple atoms, 256 atoms in total



(c) Multiple atoms, 256 atoms in total



(d) Multiple atoms, 256 atoms in total



(e) Multiple atoms, 256 atoms in total

The distribution of atom separations was measured across the database and these are plotted in figure ???. The full set is available to download from [https://github.com/BenPalmer1983/fepd\\_feru\\_potential](https://github.com/BenPalmer1983/fepd_feru_potential).

# Chapter 10

## Results: Interatomic Potential Fitting

*A computer code was developed to fit interatomic potentials to DFT calculated energy, force and stress data. It also uses bulk property data including the bulk modulus, lattice parameter, cohesive energy and elastic constants. It was written in Python and Fortran and has been designed to be easy to add new fitting data, such as cohesive energy and surface energy plots.*

*The code has been used with the results from chapter 9 to fit two potentials for FCC Fe-Pd. The cohesive energy and surface energy plots were calculated using these potentials.*

### 10.1 EAMPA Python Code

#### 10.1.1 Introduction

This code was developed to analyse interatomic potentials and fit their parameters to data. It currently has the following features:

- calculate energy, forces and stress of an ensemble of atoms
- calculate bulk properties of a potential for pure BCC or FCC structures
- fit parameters using simulated annealing, genetic algorithm, Nealder-Meade, Levenberg-Marquardt, BFGS
- 

The source code and instructions on how to use the program are available to download from GitHub.

<https://github.com/BenPalmer1983/eampa>

After several iterations, a Python-Fortran version was developed. The Python code was designed to handle the user input, output and the fitting process. The Fortran modules were designed with a separate module for each task: energy/force/stress, bulk properties, surface energy and defect calculations. The surface energy and defect module is incomplete. More modules may be easily added at a later date.

#### 10.1.2 Potentials

The code has been designed specifically for many body potentials. These include Finnis-Sinclair, embedded-atom method, two band embedded-atom method but not the modified embedded-atom method. It may be reduced to pair potentials only by setting the density function and embedding functional to “zero” (see appendix ?? for function details).

The function types available for pair, density and embedding are listed in the appendix (??). More function types may be easily added by editing the Fortran fnc module and recompiling the efs and bp modules.

Most of the functions are basic mathematical functions that use built in operators and functions. Several are more complex, such as the cubic spline, quintic spline and knot-to-knot splines.

### Cubic and Quintic Splines

These splines have been discussed in section 5.3.4. Several papers have previously used summed polynomial splines that employ the Heaviside-step function to cut off these functions at the desired points. The cubic term,  $(r - r_i)^3$ , ensures the function and it's first and second derivatives equal zero and are continuous at the cutoff points.

$$V(r) = \sum_i^N a_i(r - r_i)^3 H(r_i - r) \quad \text{where} \quad (10.1)$$

$$H(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

This was implemented in Fortran so that an efficient vectorised version could be used by Python. The Fortran code for the spline function is in the appendix ??.

### Knot to Knot Splines

A Fortran module was created for the knot-to-knot spline function. It takes a set of knots and splines between the knots pair at a time using the x, y, y' values to construct a set of linear equations to solve.

$$\begin{aligned} c_0 + c_1 x_A + c_2 x_A^2 + c_3 x_A^3 &= f(x_A) \\ 0 + c_1 + 2c_2 x_A + 3c_3 x_A^2 &= f'(x_A) \\ c_0 + c_1 x_B + c_2 x_B^2 + c_3 x_B^3 &= f(x_B) \\ 0 + c_1 + 2c_2 x_B + 3c_3 x_B^2 &= f'(x_B) \end{aligned} \quad (10.2)$$

$$\begin{bmatrix} 1 & x_A & x_A^2 & x_A^3 \\ 0 & 1 & 2x_A & 3x_A^2 \\ 1 & x_B & x_B^2 & x_B^3 \\ 0 & 1 & 2x_B & 3x_B^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} f(x_A) \\ f'(x_A) \\ f(x_B) \\ f'(x_B) \end{bmatrix} \quad (10.3)$$

A subroutine was created to calculate the coefficients for the 3rd order or 5th order polynomial to spline the two knots. This subroutine and the remainder of the module are available to view and download on the GitHub repository for this project. The Fortran code for a cubic and cubic with ZBL inner core knot-to-knot spline are included in the appendix (section ?? and section ??).

#### 10.1.3 Calculations

The calculations involved in this code centre around the calculation of energy of configurations as well as the forces and stresses when necessary. All configurations are held within the Python program, but the neighbour

list and energy (as well as stress and force) calculations are carried out in an imported Fortran module.

### Energy, Force and Stress Calculations

A potential and configuration of atoms may be loaded into the program in a number of ways. Python passes this data to the atom Fortran module. This computes the neighbour list and this is passed back to the Python program where it is stored for later use.

There are a number of functions that allow the potential as well as atom configurations (and neighbour lists) to be updated. These are handled in Python, but the calculations are carried out in the Fortran atom module, that then passes the results back to Python.

The energy, forces and stresses may be computed for a range of configurations or a single configuration. This process starts in Python, but the heavy lifting is once again completed using the Fortran module. The method used is discussed in section 5.5.3 and the subroutine used is included in the appendix (section ??).

### Geometry and Coordinate Optimisation

Once a configuration is loaded into the program, it can be relaxed using a variety of techniques. The lattice parameter may be relaxed using the built in BFGS or CG minimizers. The same functions may be used to relax the simulation cell. Where the coordinates of the atoms are relaxed, a damped molecular dynamics subroutine is called from the Fortran module that computes the forces on the atoms and moves them accordingly. This has been more successful than using built in minimizers, where no gradient data is passed, but a more promising approach might be to include the VA08 subroutine from the Harwell archives[[harwellva08](#)]. This is the same subroutine used in the Bonny potential fitting code[[114](#)].

### Bulk Property Calculations

The bulk property calculations rely on the calculation of energy of a number of configurations where specific strains

The bulk property calculations are also performed in Python using a library created with F2PY. In order to speed up the calculation, the equation of state and elastic constant configurations use the known lattice parameter set by the user. The methods used are discussed in chapter 5 and the subroutine that creates the configurations, with which the module calculates the energies used to plot the equation of state and other distortions, is included in the appendix ??.

### RSS - How Well The Potential Fits Experimental & DFT Data

The fitting process measures how well a potential fits the data when the initial parameters are given and every time a new set of parameters are proposed. They are weighted and computed in the efs and bp modules. They are the residual squared sum (RSS) of the differences between calculated energies, forces, stress, lattice parameter, cohesive energy, bulk modulus and elastic constants and the known values. The sum of these values measures how well the potential fits.

#### 10.1.4 Analytic Potential Fitting

In earlier versions, the code used tabulated potentials only. Any analytic potentials were converted to tabulated. This was for two reasons: the functions could be easily exported and used by other codes that may not have the same analytic functions, and interpolation between tabulated data was faster to compute the value and

gradient of a function at a certain point than several of the analytic functions (in particular the knot-to-knot splines).

A negative aspect is that, whilst the pair and density functions have a fixed range (from 0 to the cutoff radius), the range for the embedding functional depends on the density function, and the range of densities computed at each atom location. Rather than compute this first, I re-wrote the code to use analytic functions only, with the option of outputting as a tabulated function for use in other programs. A number of Python scripts were created to fit pre-existing tabulated potential functions into analytic functions to use as a starting point for the fitting procedure.

### 10.1.5 Overview of the Potential Fitting Process

The input files are read in by Python that then iterates over the fitting process while changing the potential. The Fortran modules calculate the energy, force and stress of the loaded configurations as well as the bulk property calculations for pure crystals of the atomic species and structures specified by the user. To save on processing time, the neighbour lists for all configurations, including those used to calculate the equation of state and elastic constants, are only calculated once.

The step by step flow of the program while fitting parameters of a potential is as follows:

1. Read input files
2. Load Fortran Modules
3. Load potentials into Python and Fortran
4. Load configurations into Python and Fortran
5. Allocate memory
6. Create neighbour list for all configurations
7. Create eos and ec configurations and neighbour lists
8. Begin fitting loop in Python
9. Loop starts - loop through potential function parameters
  - (a) Update potential and load parameters to Fortran modules
  - (b) Loop starts - loop through configurations
    - i. Calculate energy, force, stress using pre-calculated neighbour lists
  - (c) End loop
  - (d) Loop starts - bulk property calculations for each pure element
    - i. Calculate eos and ec values and bulk properties using pre-calculated neighbour lists
  - (e) End loop
  - (f) Calculate rss between these values and known/dft
  - (g) Save parameters if best
  - (h) Make new parameters
10. End loop
11. Output results

Once the fitting process has completed, the best set of parameters are stored for the user along with a plot of the potential functions, equation of state and elastic constant energy-strain distortion plots (where these are used in the fitting process).

### 10.1.6 Simulated Annealing

The simulated annealing algorithm was considered for optimisation, and may be implemented as an option in the future. The algorithm looks for better solutions and replaces the current value with better values. It also accepts worse solutions with a probability that decreases with “temperature”. Similar to the way atoms find optimal positions by annealing and cooling, the algorithm starts with a high temperature and it reduces this temperature over time. It is used to search for the global optimum.

The algorithm used is given in the appendix (section ??).

### 10.1.7 Genetic Algorithm

The global optimum is searched for using a genetic algorithm. The user chooses the parameters for the algorithm and it attempts to search for the optimum parameters to fit the potential functions to the data.

The user settings include the following:

- population size
- “fresh” (new randomly generated) parameters
- number of generations
- number of extinction events
- how often the top solutions are enhanced with linesearch and gradient descent

The subroutine that breeds the parameters randomly combines the parameters from two parent solutions to create two children parameter sets. The parents are either both from the existing pool of solutions, or a parent from the existing pool with a parent from a fresh set, and the fresh set of parameters changes with each generation (fig. ??).

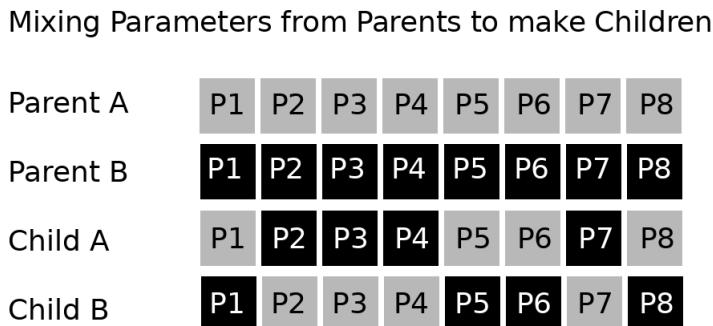


Figure 10.1: Parameter Breed Event

There is a chance that the child solutions will mutate slightly from the two parent combination, and this chance is set by the user.

During the breed event, the program will check whether or not the two children already exist within the solution pool. If they are allowed to also enter the pool, the solutions may quickly be filled with clones during the breeding process. The breed event automatically forces any clones to be varied so they become a different and distinct solution.

Every now and then an extinction event will occur that kills off any poor solutions and replaces them with variations of the better solutions. The user controls the frequency of the extinction event and the probability that a solution will be removed.

Top solutions are enhanced, at a frequency picked by the user, using a local optimisation algorithm: line-search and gradient descent. It does require many calculations and adds a substantial amount of time to the procedure, but it uses the gradient of solutions to find more optimal solutions more efficiently than randomly generated or bred solutions.

The algorithm used is given in the appendix (section ??).

### 10.1.8 Newton Gauss Optimisation

The Newton Gauss optimisation estimates the second order derivatives matrix, the Hessian, by multiplying the Jacobian and its transpose. It is not yet implemented in the code, but may be in the future. The algorithm is used to find a local optimum, but it can be unstable or return errors.

$$\begin{aligned} \vec{J}^T \vec{J} \vec{p} &= \vec{J}^T \vec{r} \\ \vec{r} &= \vec{y} - f(\vec{x}, \vec{p}) \end{aligned} \tag{10.4}$$

### 10.1.9 Levenberg-Marquardt Optimisation

The Levenberg-Marquardt algorithm is similar to the Newton-Gauss method, but the estimate for the Hessian does not need to be calculated for each loop of the algorithm. The parameter,  $\lambda$ , is changed and the Hessian is updated periodically. This is not yet implemented in the code, but may be added in the future.

$$\begin{aligned} \left( \vec{J}^T \vec{W} \vec{J} + \lambda \text{diag} \left( \vec{J}^T \vec{W} \vec{J} \right) \right) \vec{p} &= \left( \vec{J}^T \vec{W} \vec{r} \right) \\ \vec{r} &= \vec{y} - f(\vec{x}, \vec{p}) \end{aligned} \tag{10.5}$$

### 10.1.10 Python SciPy Optimization Functions

Additional optimisation functions have been used by importing the SciPy module into the program, and these may be used in conjunction with those listed above. Several have been used including the bin-hopping, Nealder-Meade and BFGS. The bin-hopping was unreliable and crashed on a number of occasions, so I used the simulated annealing and genetic algorithms programmed specifically for this work. The Nealder-Meade and BFGS were both used following the run of global optimisation and increased the efficiency of improving the parameters.

### 10.1.11 Minimising Runtime

#### Python, Fortran & OpenMP

When fitting a potential, for each trial set of functions, the program must calculate the energy, forces and stress for each of the configurations and tens or hundreds of energy only calculations for the equation of state and elastic constant calculations. Python is a well used programming language in many areas of Science and it has the benefits of many modern languages. For certain specific tasks, it is slow in comparison to Fortran.

The energy, stress and force calculations were programmed in Fortran and use OpenMP to calculate the many configurations in parallel. On a single thread, the fitting process could take weeks, but on a 20 core node the calculation time was reduced to under a couple of days.

### Function Caching

not relevant

The first version of the code used tabulated potentials, and this removed the task of programming different functions (and the wide range of complexities amongst those functions) and replaced it with a quick search and interpolation of tabulated data. However, the final version was developed to use and fit with analytic potentials. Several of the optional functions, including the knot-to-knot spline function, drastically increased the computational time.

The bulk property module in particular uses configurations with atoms in regular places, with there being a small set of unique atom separation values. Rather than generate tabulated data for each function, to then interpolate, the input and output values are cached. Once the value is calculated, it is quickly retrieved for subsequent calculations with the same input value. This has increased computation speed of the analytic potentials to a speed similar to using tabulated functions.

#### 10.1.12 Interpolation for Non Analytic Potentials

In the current version, only analytic potentials are available for fitting. There are several subroutines ready if tabulated functions are reintroduced. These subroutines use Lagrange polynomials to interpolate values of the function (eq. ?? ??).

$$D = \{(x_0, y_0) (x_1, y_1) (x_2, y_2) \dots (x_n, y_n)\} \quad (10.6)$$

$$p_n(x) = \sum_{i=0}^n y_i L(i, x) \text{ where } l(i, x) = \prod_{j=0, j \neq i}^n (x - x_j) \quad (10.7)$$

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 (eq. ??).

$$q_n(x) = \sum_{i=0}^{i=n} y_i g(i, x) \quad (10.8)$$

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

A pseudocode for both interpolation of the function points and derivatives are available in the appendix (section ?? and ??).

### 10.1.13 Effective Gauge Transforms

In order to create a potential for an alloy an effective gauge transform, as described in section 5.4.4, was applied to each pure potential. A mode added to the EAMPA program reads in the analytic potential and creates a numeric potential that has had the transform applied. This allows the pure potentials to be used together, although a cross pair potential is still needed.

## 10.2 Iron-Palladium Potential

### 10.2.1 Reference Database

The database of reference configurations discussed in chapter 9 was used to derive this potential. It was designed for FCC Fe rather than BCC Fe, and this is one point that distinguishes this potential from those that already exist.

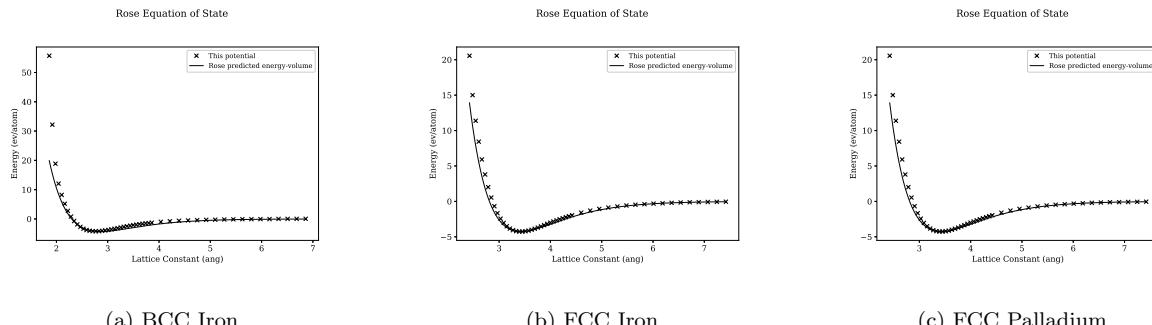
During the fitting process, there was a desire to have a reasonably good fit to bulk property values as well as the defect configurations, perturbed atom configurations and so on. This was quite a challenge to fit to so many variables and as a result more weight was given to the bulk and slab/surface configurations.

There were a number of options available when choosing certain parameters, such as using DFT to compute cohesive energies based on relaxed and isolated configurations, or to extrapolate based on known energies. The choices made are listed below and these apply to both the Fe-Pd and Fe-Ru potential.

- all cohesive energies were computed based on known experimental values and the DFT computed relaxed energy of those structures
- all bulk property values are those computed using DFT

### 10.2.2 Potential

The full potential is available from [https://github.com/BenPalmer1983/fepd\\_feru\\_potential](https://github.com/BenPalmer1983/fepd_feru_potential) and is ready to use with LAMMPS.



(b) FCC Iron

(c) FCC Palladium

(a) BCC Iron

# Chapter 11

## Conclusions

*It is known that Cr provides protection to stainless steel through a passive protective layer, but this becomes ineffective to protect the steel under irradiation, as Cr is depleted from the grain boundary. Platinum group metals also provide a mechanism to protect steel from corrosion, and experimentation or simulations may be used to investigate whether they can maintain high enough concentrations at the surface to continue to protect the steel against corrosion under irradiation. If either of the two chosen PGMs, Pd or Ru, are depleted at the grain boundary under irradiation as well as depletion of Cr, the steel will become susceptible to corrosion.*

*Experimental testing is cheaper, more controllable and more rapid at causing damage in a targeted volume if an ion beam is used rather than a neutron source. Either source will cause the target material to become radioactive. It is important to be able to calculate how radioactive, and how long after irradiation (to the desired damage, in DPA) it will be until the material is safe to be handled.*

*Computer simulations to simulate radiation damage will be much too large and complex for first-principles calculations. They will need either classical molecular dynamics or kinetic monte carlo simulations to encapsulate the much larger simulation volumes, to contain the atoms effected by the damage cascade, and to represent the grain boundary. This requires derivation of an appropriate interatomic potential using experimental data and first-principles data where it is not available, or is either impossible or impractical to measure.*

### 11.1 Inter Granular Stress Corrosion Cracking

Austenitic stainless steel is particularly susceptible to inter granular stress corrosion cracking. It has good properties, including a good resistance to corrosion due to its high Cr and Ni content. Under irradiation, the Cr at grain boundaries is depleted with the formation of Cr carbides. Three requirements for inter granular stress corrosion cracking to occur are:

- a susceptible material (austenitic stainless steel)
- stress, welds, swelling, pressure, any applied stress (high pressure in reactor environment, swelling due to irradiation)
- a corrosive environment (radiolysis of water)

The passive Cr layer, under normal conditions, protects the steel from corrosion. The steel is made up from small grains, and the surface of these grains of crystalline metal are where the protective layer covers. During irradiation, Cr is depleted, forming Cr carbides between the grain boundaries. As the percentage of Cr drops, the surface becomes prone to corrosion.

Small quantities of platinum group metals may be added to a steel to also increase the corrosion resistance of the steel. Platinum group metals are rare and much more costly than Fe, Ni, Cr. Where irradiation is not present, it has been shown experimentally that the benefit due to Pd is lost due to the formation of Pd-Mn nanoparticles, where Mn is present in the steel, although no such particles are present when Pd was replaced by Ru. Pd does provide corrosion protection via cathodic modification when Mn is reduced to very small amounts as it no longer forms Pd-Mn precipitates during sensitisation. The outstanding question is whether or not Pd or Ru are depleted, enriched or are unaffected at the grain boundary when under irradiation.

## 11.2 Predicting Ion Induced Radioactivity

Testing a material inside a nuclear reactor is an expensive experiment. There are many more ion sources around the world than high flux neutron sources, and ion beams are easier to direct due to the charge of the ion versus the neutral charge of the neutron. The ion energies required to create similar sized damage cascades in the material are high enough to also have a chance of transmuting the target nuclei. These transmuted nuclei are most likely going to be radioactive. To reach damage dose levels comparable to a component in a Gen IV reactor by the end of its life (up to 200 DPA), the target material will become dangerously radioactive.

Bateman's equation may be used to calculate the amount of an isotope in a decay chain of several isotopes, and thus how radioactive an isotope would be after a certain time. Major modifications were required to be made to the equation to also include branching factors and source rates for isotopes generated by an ion beam.

### 11.2.1 Modified Equation

Whilst the equation was first tackled by solving the differential equations head on, it became obvious that using Laplace transforms would be the logical choice; this was also the route Bateman followed when deriving the original equations. Once transformed, a numerical algorithm (Gaver-Stehfest) was programmed and used to solve, with some success.

The numeric method is not an exact solution, and the errors incurred often resulted in negative amounts of an isotope, and this was unacceptable. More progress was made towards solving the problem analytically, and by using partial fractions, obviating the need for a numerical method.

The analytic solution was tested against a simple numeric solver and the results were in good agreement between the modified equation and numeric solutions (appendix ??).

### 11.2.2 Activity Code

The Activity code was created to automate the process of calculating the activity of an ion irradiated target. The program:

- reads in simulation details (target composition, thickness, duration etc)
- reads in ion cross section data from the TENDL database
- processes SRIM exyz data points
- calculates the amount of each isotope by the end of the simulation

- calculates the activity of each isotope and the dose that an average human would be exposed to

A number of simplifications are made to the model but it predicted the radioactivity of certain peaks reasonably well with a sample of proton irradiated pure Fe. Given more time a range of alloys would also be irradiated by a proton beam and, once their activity is measured at several time intervals after irradiation, they would be compared to the activity predicted by the Activity code.

The first version of the Activity code was replaced by a second version that used Python rather than Fortran. The first version was more cumbersome for new users and it crashed for certain isotope, and these problems were fixed in the new version.

### 11.2.3 Iron Irradiation

During the experimental stage of this work it was determined that the isotope to focus on would be Cobalt-55 and the associated 931keV peak it generates would be the peak to measure. The equipment was calibrated specifically for the purpose of measuring this gamma. If this is the case then the value predicted by the Activity V2 computer code is in good agreement (table 7.5).

However, the second version of the Activity code collates and displays data more clearly than the first version. It is clear that there should also be a nearby 935KeV gamma from the decay of Manganese-52. The sum of these nearby gammas is predicted to be almost 120kBq and, if both peaks were tallied in the experimental work due to the energy resolution of the detector, the predicted value is three times that of the measured value of 44kBq.

It is quite possible that the code is giving a value three times larger than expected and reasons for this will be discussed below.

### 11.2.4 Molybdenum Irradiation

Similar to the results discussed for Iron, the activity results computed for Molybdenum have positive and negative attributes. There are gaps in the available cross section data, but the most prominent gammas are predicted by the computer code. The radioactivity computed is too high, but for each of the five peaks measured by experiment, the values are out by a similar factor.

### 11.2.5 Beam Flux Reduction and Gamma Attenuation

The cross section data used in the second version has been checked for a number of cases against EXFOR data and is in good agreement with this. Certain data for reactions is missing in particular those of observationally stable and isotopes with extremely long isotopes.

Thin foils are ideal to compare experimental results to the data files and activity code as ions will not lose a great deal of energy travelling through such a shallow target. However, the much thicker targets used here for Iron and Molybdenum reduce ion energies significantly if not completely.

It could be argued that as ions pass through the target the energy and intensity of the beam decreases, and this is not modelled with the activity code. By imagining the target as thin laminae perpendicular to the beam it is easy to consider that the ions will enter the first layer and lose those that react with target atoms. As the beam enters the next lamina it will have a lower energy and a lower intensity.

The change in energy is accounted for by virtue of the SRIM exyz ion trajectory file but how the beam intensity decreases is not. How many ions are captured through nuclear reactions and is it a significant percentage of the total number of ions? Particles and residuals were calculated to be produced at a rate of approximately  $6 \times 10^{10}$

per second for the Iron target and  $7 \times 10^{10}$  per second for the thick Molybdenum target. The beam rates were  $3 \times 10^{12}$  and  $3 \times 10^{13}$  protons per second respectively and so, even if the estimate that a proton was removed for each reaction the beams would only lose 2% and 0.3% of their intensity due to transmutation reactions.

Another possibility for the over estimate in prediction is the attenuation of gammas travelling through the target material to the detector. This was discussed in the work of my colleague[175] and the inhomogeneous distribution of radioactive material in the Molybdenum target was discussed in this work (section 7.8.2). Calculations using the NIST attenuation coefficient for Molybdenum show that 500keV gammas have their intensity reduced to 91% through a 1.0cm target. The intensity of 1MeV gammas is reduced to just 94% through the same target.

The reduction through loss of ions due to reactions and attenuation of gammas does not explain the increase in predicted activity. There may be another reason behind this, or there may be issues with the experimental work.

### 11.2.6 More Experimental Data Required

I have not had a great deal of experience practically measuring gammas, and I would expect there to be errors introduced due to my own inexperience. The range of beam energies, elements and beam duration are also very restricted.

In past experimental work discussed in section 4.4.3, thin foils were used to measure reaction probabilities, and this data influences models such as TALYS. In some two out of the three examples reviewed, the data beam energy was degraded using a stack of foils perpendicular to the beam, with those closest to the aperture receiving the highest energy protons. It is hard to imagine whether foils stacked in this way would be much different to a solid target of the same thickness, but this should be tested experimentally.

Ideally a range of target materials of varying thicknesses would be irradiated with protons of energies ranging up to the limit of the cyclotron, perhaps 60MeV, and the Gamma spectra of each measured at set times after irradiation. This data would be compared to the gamma lines predicted by the activity code to help give a better understanding of the accuracy of the code.

### 11.2.7 100DPA Activity Predictions

Using dose rates calculated with SRIM, the code was used to predict how radioactive Fe would become at a dose rate of 100DPA. It assumes the beam source is the Scanditronix MC40 using protons at maximum current for an uninterrupted duration of 134-391 days. In the worst case scenario, 25MeV protons result in over 20 grays per hour if irradiated to the required damage dose. However, by reducing the beam energy to 15MeV radioactivity of the target is reduced by a factor of 1,000. It may be reduced further by lowering the energy to 5 or 10MeV, but for a 0.5mm piece of steel, the ions would not have enough energy to pass through and the damage would be disproportionately concentrated in the top 100-250 microns.

### 11.2.8 Original Contribution

The modified Bateman equation as derived is an original contribution to knowledge. The activity code is also an original contribution, and its use shows how adjusting beam parameters will reduce the amount of radioactive material created and the risk to those handling irradiated samples.

## 11.3 Iron-Palladium Potential

### 11.3.1 Restricting the Potential to Iron-Palladium

It would be too complex a task to develop an interatomic potential that includes Fe, Pd, Cr, Ni and any other elements that make up a typical sample of austenitic stainless steel; Fe and Pd are the starting point, with the possibility of adding more elements to the potential in the future.

Pd exists in the FCC state at normal conditions, but Fe is BCC. The relatively high percentage of Ni in the steel stabilises the austenitic FCC structure. The bulk properties, such as the lattice parameter, bulk modulus, elastic constants and so forth, are experimentally known for Pd. As Fe doesn't exist in the gamma phase at normal conditions, DFT calculations were used to find the bulk properties of gamma Fe.

### 11.3.2 Collinear Spin DFT

With most models, there will be a trade off between simplicity, accuracy and the computational cost of the model. The parameters that had the largest impact on how long a DFT calculation will take to compute are the planewave energy cutoff and number of k-points used to integrate the Brillouin zone. These should be reduced as much as possible, while keeping the energy and force results within a tolerance determined by the user.

A choice in the complexity of the calculation also had to be decided upon, and this was whether to use a non-spin, collinear spin or non-collinear spin calculation. Pure Fe under normal conditions has a BCC structure and energetically favours a ferromagnetic configuration. Cr also has a BCC structure under normal conditions, but it's optimum structure magnetically is antiferromagnetic. Gamma phase Fe (FCC) is also antiferromagnetic, and so it will be important to include magnetism in the DFT calculations.

The options in increasing complexity, and computational time, are:

- non-magnetic
- collinear - suitable for high symmetry, ferromagnetic/antiferromagnetic
- non-collinear - spin may not be aligned in the same direction

For the Fe and Pd calculations, collinear spin was used and the starting configurations of the atoms were set up in accordance with the known configurations i.e. BCC Fe was configured such that the spins were aligned in the same direction, and FCC were set up in an antiferromagnetic setting. The collinear DFT calculations showed that Fe FCC, where magnetism is taken into account, is slightly tetragonal with the atoms arranged in a FCT structure.

A Python (QECONVERGE) code was developed to automatically converge the aformentioned parameters. Whilst automation worked well for the planewave cutoff parameters the choice of smearing type, the amount of smearing and number k-points was more difficult to automate. It was more useful to select a range of k-point values and smearing values and produce 2D force and energy plots so the user could make a decision based on these and choose the parameters that best suit their requirements.

During testing, a much larger number of k-points were needed for the predicted properties of Aluminium to replicate the known properties reasonably well. Unfortunately, the number of k-points used for Fe and Pd was constrained by the amount of time and the RAM available on the compute nodes. Ideally, more k-points would have been used, but this was not possible.

### 11.3.3 Bulk Properties of Orthorhombic FCC Iron

As pure Fe does not exist in its gamma phase at room temperature, DFT calculations were used in place of experimental data. As eluded to in the previous section, collinear spin DFT was selected. Although a smaller number of k-points were used, the calculation lasted for approximately 30 days from start to end.

Rather than calculate the input files manually, a program was created to automate this process (QE\_EOS). It was designed to cache input and output files for the DFT program, PWscf, such that it would load an output file from the cache if it had already successfully run, to reduce the calculation time.

The relaxed (energy minimised) shape of the antiferromagnetic FCC Fe was not cubic, but orthorhombic. The equation of state calculation is specifically for a cubic crystal, but the elastic constants were calculated for the orthorhombic crystal. From the 9 calculated elastic constants the melting point, bulk modulus, shear modulus and Young's modulus were calculated (section 9.6.5 and appendix ??). The resulting structure was stable and were well within a magnitude of the known values for BCC Fe. These data are an original contribution to science.

### 11.3.4 Palladium-Iron Configurations as Fitting Data

To be able to fit the Fe-Pd potentials, the bulk properties of both pure FCC Fe and FCC Pd were either calculated or taken from experimental values. The potentials were then trained to fit this data. As well as this, a number of configurations of Fe, Pd and Fe-Pd were generated with their atoms slightly displaced from their lattice positions. This would give a wider range of atom separations and force values for the fitting program to use.

The force data were entirely from first-principles calculations, and ideally there would have been more configurations and computed with a higher number of k-points. In addition, a wider range of vacancies, interstitials and mixing concentrations would have been desirable. The atomic percentage on Pd or Ru doped is half a percent which would be better represented with a one PGM atom in a 4x4x4 FCC 256 atom supercell rather than the 32 atom cells used, but again this was a constraint imposed by computational resources.

### 11.3.5 Iron-Palladium Potential

The first derived Fe-Pd potential reproduces the properties of the material with a minimum error of 0.0% for the cohesive energy for both Fe and Pd, and a maximum error of 8% for the bulk modulus of Fe.

The cohesive energy and surface energy plots have a number of irregular bumps (fig ??), and it is hard to say whether or not these are correct without also computing them using DFT.

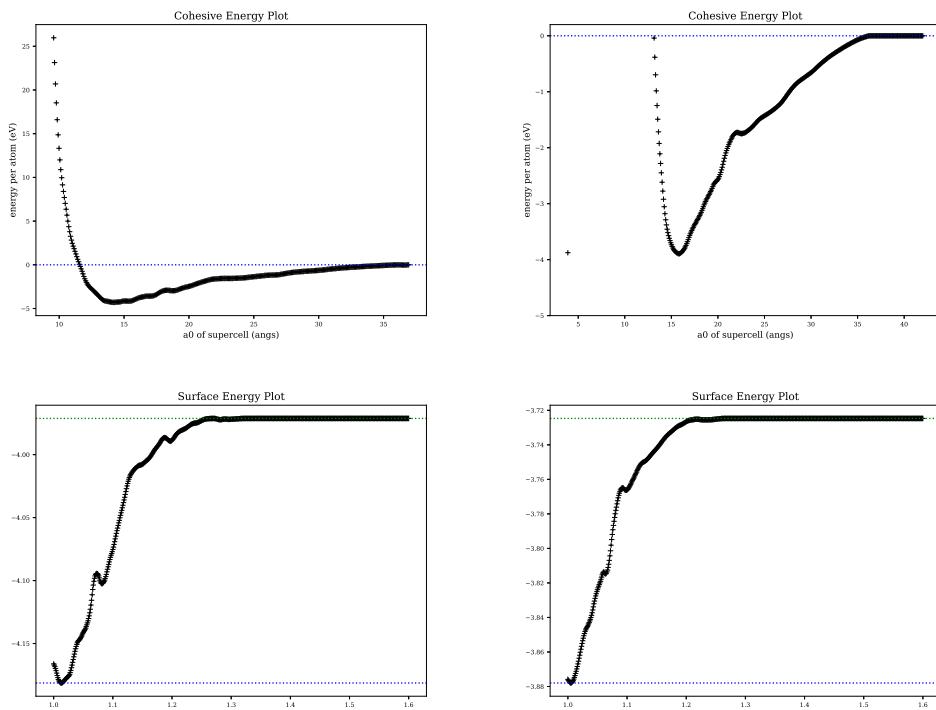


Figure 11.1: Cohesive energy and surface energy plots - detailed plots in the appendix ?? and ??

The second derived Fe-Pd potential reproduces the properties of the material with a minimum error of 0.0% for the cohesive energy of Pd and the  $C_{22}$  elastic constant for Fe, and a maximum error of 4.8% for the bulk modulus of Fe.

Despite having a better fit to the properties of the material used, the cohesive energy and surface energy plots are less well behaved than those in the first version (fig. ??). The optimisation algorithms blindly fit potentials to the data available, and it is apparent that more DFT computed energies are required for a variety of configurations, including splitting the bulk into two surfaces and expanding atoms from bulk into isolated atoms.

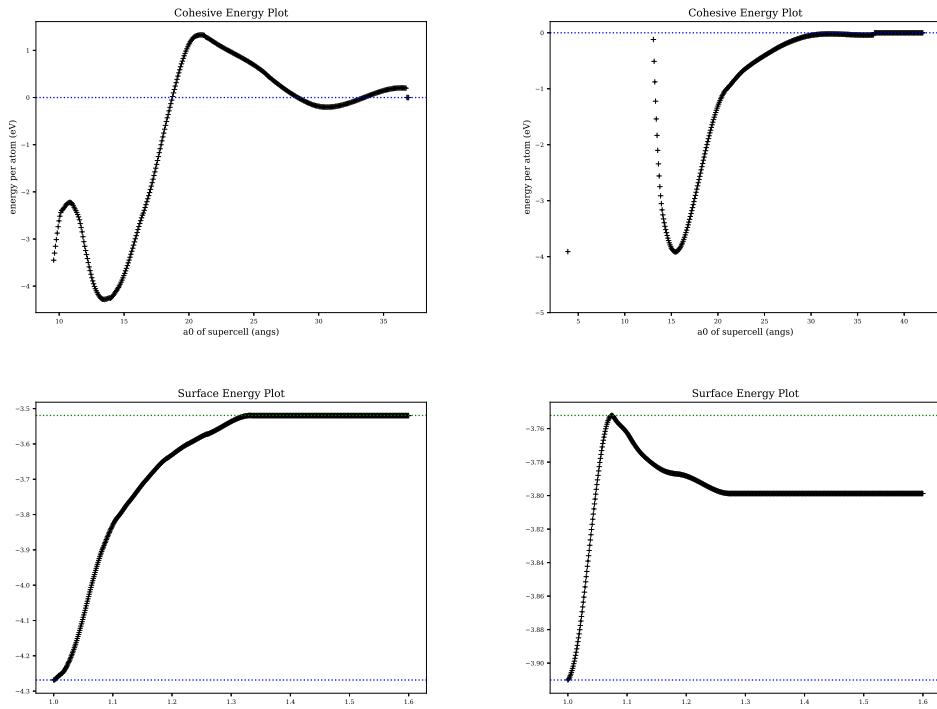


Figure 11.2: Cohesive energy and surface energy plots - detailed plots in the appendix ?? and ??

The non-cubic geometry of antiferromagnetic FCC Fe also suggests that the energy surface should be considered in all three planes, both with DFT and in the fitting process.

### 11.3.6 Original Contribution

The fitting code is an original contribution as it allows the fitting of metal potentials in the form of the embedded-atom method and two band embedded-atom method to DFT data and bulk properties. Other similar codes do exist, such as FitPot and PotFit, but at the time of developing my code the PotFit did not have the ability to fit analytic potentials (although, this feature is now available) and does not have, in its available form, the ability to fit the potential to the bulk properties of the material (although the Sheng version did have this ability). The code developed here may be modified in the future to add surface energy and point defect energies as data to fit to. It would require the addition of a geometry relaxation subroutine to optimise the atom locations as the basis vectors change.

## 11.4 Two-Band EAM Contribution to DL-Poly

During the early stages of this work it was clear that the Embedded Atom Method was a good candidate for the type of potential to be derived as it was well suited to modelling metals and had been used previously (along with the similar FS potential) to model radiation damage[25]. Work by Prof. Ackland at the University of Edinburgh considered using a separate density function and embedding functional to represent the S-Band and D-Band of transition elements such as Caesium. Work by P. Olsson and J. Wallenius also used this two band version to model Fe-Cr with a later Fe-Cr potential being developed by Bonny et al to replicate the mixing enthalpy as a function of Cr content.

After meeting with Prof. Todorov at Daresbury Laboratory, the source code for DL-Poly was modified to include new keywords, additional arrays to store the two-band density and embedding data for the second band

and modifications to the energy and force subroutines. This was then released with version 4.05 of DL\_POLY in July 2013.

Ultimately, only EAM potentials were fit in this work. In the future, as more elements are added to the potential, in particular Cr, the 2BEAM may be required. The molecular dynamics code and fitting code both have the capability to use this type of potential.

# Chapter 12

## Future Work

*At a certain point, work on various codes developed in this work, as well as the resulting Fe-Pd potential(s) had to stop, but there are a number of improvements that could be made given more time.*

### 12.1 Activity Code

#### 12.1.1 Experimental Activity Readings for Ion Irradiated Targets

The Activity code was developed using data from the TENDL data file. To test the validity of the code, an Iron sample was irradiated and its activity was measured several days after irradiation.

To validate the code further, a range of pure targets and alloys would be irradiated by a cyclotron, with targets of varying thickness and beams of varying fluences and energies. This would generate data to be used to compare to the results predicted by the code.

#### 12.1.2 Code Improvements

Ideally, a built in ion transport code similar to SRIM would be developed. Failing this, an improvement in the reaction rate calculation and a rewrite of the code in Python with F2PY to improve ease of developing in the future.

Given the variation of energy ranges between cross section databases, it would be useful to give the user a selection of a variety of TENDL and similar data files to choose from. In addition to this, reaction cross sections could be computed over a set energy range using the TALYS code specifically for use in the Activity program.

Finally, extending the range of projectiles in the Activity code to include Deuterons, Tritons and Helium would be useful as the cyclotron is capable of accelerating these particles as well as protons. It would require an extension of the database to include the relevant cross section files.

### 12.2 Potential Fitting and MD Simulations

Two potentials were derived for Fe-Pd that reproduced the bulk properties of each element individually reasonably well. Cohesive energy and surface energy plots could be improved by adding new modules to help fit the potential, and overall more DFT data could be collected.

### 12.2.1 Larger Supercells

In many examples in the literature, the supercell sizes were at least 4x4x4 containing 256 atoms for FCC crystals. These require much more memory and processing power, especially considering the lack of symmetry and inclusion of magnetism in the calculations. More calculations over a greater range of randomised atom locations, with a varying percentage of palladium (1 atom per 256, 2 atoms per 256, 3 atoms per 256 and so on) would also help towards fitting a better potential.

### 12.2.2 Properties of Fe-Pd Alloy

The properties of palladium were known and those of pure FCC iron were calculated. Using DFT to calculate the properties of a 1-2% iron-palladium alloy would provide another set of data to fit the potentials to.

### 12.2.3 Improved Fitting: Surface Energy, Cohesive Energy and Defects

The cohesive energy and surface energy plots for the derived iron and palladium potentials were recognisable, but they were not smooth and did have many small bumps in the plots. Having plotted the same for Aluminium, the DFT results were much smoother.

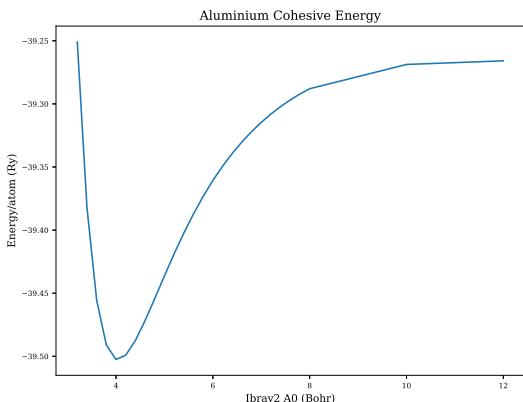


Figure 12.1: DFT Aluminium Cohesive Energy

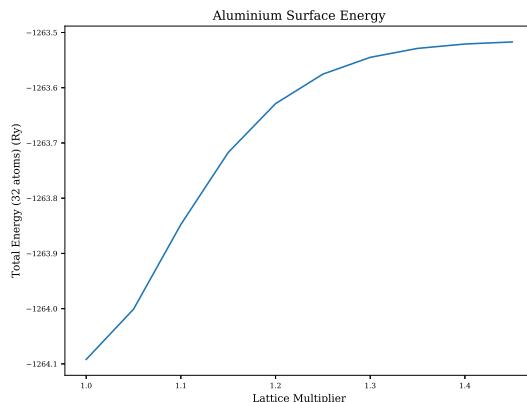


Figure 12.2: DFT Aluminium Surface Energy

Rather than just fit the data to the energy surface, it would be useful to create DFT configurations for the cohesive energy and surface plots for both Iron and Palladium. These configurations, and the energies, forces and stresses computed by DFT would be used to improve the fitting process.

Currently the fitting code has two modules: EFS, to compute energy, forces and stress of a set of atom configurations provided and BP, to calculate the material properties (bulk modulus, elastic constants etc). A new module could be created to calculate the value of the surface energy. It would require computing the relaxed energy of the potential for the bulk material. Next a slab, with enough vacuum along its surface(s) to prevent interference over the periodic boundary condition, would have the atomic positions relaxed and the energy (for the potential) calculated. This would allow computation of the value of the surface energy to be used to improve the potential.

The computation of the cohesive energy could also be implemented into a module. The relaxed bulk energy would be computed as well as that of isolated atoms, and this would give enough data to compute the cohesive energy.

Finally, a defects module could be created. Common defects, and their experimental or DFT computed energies, would be provided by the user. The module would create the configurations based on the user's input. The

volume and positions of the atoms would need to be relaxed with the current iteration of the potential for each defect, and the energy computed. This, with the energy in bulk, would compute the energy of each defect.

#### 12.2.4 Improved Potential: Non-Isotropic Structure

With a collinear spin-polarized DFT calculation, the relaxed structure was slightly longer in the y direction. Investigating the use of an angularly dependent potential might be worthwhile and the MEAM type potential with an angularly dependent electron density would be a starting point (section 5.3.3).

#### 12.2.5 Improved Potential: Fe-Ni-Cr-Pd

Given more time and a larger computer, more complicated DFT calculations could be performed that better represent the material, austenitic stainless steel. 1% palladium doped steel could be represented as a 4x4x4 supercell containing 256 atoms. For 304SS approximately 50 chromium, 25 nickel, 178 iron and 3 palladium atoms would be needed (and small variations on these figures).

# Acronyms

**2BEAM** two band embedded-atom method. 73, 89, 169

**ABWR** Advanced Boiling Water Reactor. 4, 7

**AGR** Advanced Gas Reactor. 3–5, 8, 9, 38

**ASME** American Society of Engineers. 29, 30

**AVF** Azimuthally Varying Field. 49

**BCC** body centered cubic. 14, 28–30, 36, 63, 64, 73, 74, 80, 104, 119, 155, 174, 178, 179

**BFGS** Broyden Fletcher Goldfarb Shanno. 57, 92, 95, 97, 168

**BWR** Boiling Water Reactor. 3, 11, 31, 38

**BZ** Brillouin zone. 104, 114, 115

**CANDU** Canada Deuterium Uranium. 3

**CCGT** Combined Cycle Gas Turbine. 6, 10, 11

**CERN** Conseil Européen pour la Recherche Nucléaire. 43

**CG** Conjugate Gradient. 95

**CPH** face centered tetragonal. 119

**CPU** central processing unit. 98

**DFT** Density Functional Theory. i, 16, 58, 64, 66, 83, 85, 91, 92, 98, 99, 110, 112–116, 153, 155, 158, 159, 161, 167, 170–172, 175, 177–179

**DNA** Deoxyribonucleic Acid. 17

**DPA** displacements per atom. i, 14, 31, 33, 34, 121, 129, 146

**EAM** embedded-atom method. 15, 16, 57, 72–74, 80, 82, 89, 166, 168

**EBR** Experimental Breeder Reactor. 33

**ECIS** Equations Couplées Itérations Séquentielle. 50

**ECP** electrochemical corrosion potential. 38, 39

**ELSY** European Lead Fast Reactor. 9

**ENDF** Evaluated Nuclear Data File. 52, 54

**EPR** European Pressurised Reactor. 7

**EXFOR** EXchange FORmat. 49, 139, 140, 148–150

**EXYZ** Energy, x, y, z. 127

**FCC** face centered cubic. i, 28–30, 36, 57, 63, 64, 67, 73, 104, 119, 159, 160, 170, 177–180

**FCT** face centered tetragonal. 160, 177

**FS** Finnis-Sinclair. 16, 57

**Gen II** Generation II. 3, 4, 31, 33

**Gen II+** Generation II+. 4

**Gen III** Generation III. 4

**Gen III+** Generation III+. i, 1, 4, 7, 13

**Gen IV** Generation IV. i, 1, 2, 4, 8, 9, 11, 13, 31, 39, 46, 121, 145

**GFR** Gas-Cooled Fast Reactor. 11, 39

**GGA** generalized gradient approximation. 112–115, 171, 172, 178

**GPU** graphical processing unit. 83

**HCP** hexagonal close packed. 170, 180

**HFIR** High Flux Isotope Reactor. 43, 44, 46

**HK** Hohenberg-Kohn. 105–107, 114

**HRB** Hardness Rockwell B. 30

**IAEA** International Atomic Energy Agency. 49

**IASCC** irradiation assisted stress corrosion cracking. 36

**IBZ** irreducible Brillouin zone. 104, 114

**IGSCC** inter granular stress corrosion cracking. 1, 28, 30, 34, 36, 37, 39

**IKE** inverse Kirkendall effect. 33

**INS** Institute for Nuclear Study. 49

**ITER** International Thermonuclear Experimental Reactor. 12

**JEFF** Joint Evaluated Fission and Fusion File. 53, 54

**KE** Kirkendall effect. 33

**KS** Kohn-Sham. 111

**LAMMPS** Large-scale Atomic/Molecular Massively Parallel Simulator. 77, 83, 87, 88, 91

**LDA** local density approximation. 111, 112, 171, 178

**LFR** Lead-Cooled Fast Reactor. 9, 11

- LHC** Large Hadron Collider. 43
- LINAC** linear accelerator. 42, 45
- LM-BFGS** Limited-memory Broyden Fletcher Goldfarb Shanno. 95
- LMA** Levenberg-Marquardt Algorithm. 92, 96, 97
- LSDA** local spin density approximation. 111–115, 171, 172
- LWR** Light Water Reactor. 38
- MCNP** Monte Carlo N-Particle. 19
- MD** molecular dynamics. i, 58, 80, 83, 87, 91, 153
- MEAM** modified embedded-atom method. 76, 166
- MOX** mixed oxide. 84
- MSR** Molten Salt Reactor. 12
- NAA** neutron activation analysis. 57
- NBS** National Bureau of Standards. 44
- NBSR** National Bureau of Standards Reactor. 45
- NDS** Nuclear Data Services. 48
- NEA** Nuclear Energy Agency. 53
- NG** Newton-Gauss. 95, 96
- ORNL** Oak Ridge National Laboratory. 43, 45
- PBE** Perdew-Burke-Ernzerhof. 112, 113
- PBESOL** Perdew-Burke-Ernzerhof functional revised for solids. 112, 113
- PBR** Pebble-Bed Reactor. 10
- PGM** platinum group metal. i, 1, 39–41, 160, 180
- PKA** primary knock-on atom. 13–15, 26, 27, 31, 33, 80, 83–85
- PWR** Pressurised Water Reactor. 3–5, 7, 11, 12, 31, 38, 39
- PZ** Perdew-Zunger. 113
- QP** Quadratic Programming. 95
- RBMK** Reaktor Bolshoy Moshchnosti Kanalnyy. 3
- RIS** radiation induced segregation. 33, 34, 39
- RSS** residual squared sum. 178, 179
- SA** Simulated Annealing. 95
- SCC** stress corrosion cracking. 33, 38, 40

**SCF** self consistent field. 108, 116, 160–162, 164, 165, 175

**SCWR** Supercritical-Water-Cooled Reactor. 9, 11, 12

**SEM** scanning electron microscope. 57

**SFR** Sodium-Cooled Fast Reactor. 9, 11, 12, 39

**SIMS** secondary-ion mass spectrometry. 57

**SLAC** Stanford Linear Accelerator Center. 42

**SNS** Spallation Neutron Source. 42, 45, 46

**SRIM** Stopping Range In Matter. 42, 48, 54, 56, 71, 121, 127, 129, 146

**TEM** transmission electron microscope. 40, 57

**TENDL** TALYS-based Evaluated Nuclear Data Library. 41, 42, 48, 52–54, 135, 138, 140, 141, 149

**TGSCC** trans granular stress corrosion cracking. 37

**TISE** Time Independent Schrödinger Equation. 98, 100–102, 105, 110, 113

**TRIM** TRansport In Matter. 54–56

**TRISO** Tristructural Isotropic. 12

**TRIUMF** TRI-University Meson Facility. 42

**VHTR** Very-High-Temperature Reactor. 10, 11

**VPI** vacancies per ion. 146

**VVER** Vodo-Vodyanoi Energetichesky Reaktor. 3

**XC** exchange-correlation. 110, 111, 113

**ZBL** Ziegler-Biersack-Littmark. 15, 71, 72, 79, 80, 167, 168

# Glossary

**304SS** Austenitic stainless steel: 18-20%Cr, 8-11%Ni. 30, 33, 35, 39, 40

**316SS** Austenitic stainless steel: 16.5-18.5%Cr, 10-13%Ni, 2.0-2.5%Mo. 30, 36, 39

**allotrope** an element that exists in multiple forms/structures - the common example of allotropes are graphite (hexagonal sheets) and diamond (regular tetrahedral structure) for carbon. 29

**antiferromagnetic** neighbouring electrons orientate in opposite directions, cancelling out overall magnetism. 76

**barn** Unit for cross sectional area ( $1.0 \times 10^{28} m^2$ ). 21

**Becquerel** Becquerel (Bq) - decays per second. 22

**Bloch theorem** electron wave functions may be written as a sum of plane waves  $\psi_{\vec{k},n}\vec{r} = \sum c_{\vec{k}+\vec{G},n} \exp(i(\vec{k} + \vec{G})\vec{r})$ . 104

**Bohr magneton** natural unit for magnetic moment  $5.788 \times 10^5 eV T^{-1}$ . 177

**Brillouin Zone** the Wigner-Seitz cell in reciprocal space. 104

**C** Carbon  $Z = 6$ ,  $A_r = 12.00$ . 28

**Cauchy pressure**  $C_{12} - C_{44}$ . 76

**Co** Cobalt  $Z = 27$ ,  $A_r = 58.93$ . 1

**Cr** Chromium  $Z = 24$ ,  $A_r = 52.00$ . 1, 2

**Cu** Copper  $Z = 29$ ,  $A_r = 63.55$ . 33

**Curie** Curie (Ci) - activity of 1 gram of Radium-226 -  $1\text{Ci} = 3.7 \times 10^{10}$  decays per second. 22

**enthalpy** sum of internal energy plus pressure times volume  $H = E_{internal} + PV$ . 75

**eutectic** A eutectic contains two or more elements, and has a lower melting point than any of its constituent elements.. 9

**Fe** Iron  $Z = 26$ ,  $A_r = 55.85$ . 1

**Fermi energy** the energy of the highest occupied state. 104

**fermion** half-integer spin particle - includes electrons, protons, neutrons. 109

**fissile** Fissile materials can undergo a nuclear chain reaction with slow (thermal) neutrons. 2

**Gray** Gray (Gy) how much energy is deposited into matter and 1Gy is equal to 1 Joule per kg. 22

**Hamiltonian** the total energy of a system - an operator in quantum mechanics,  $\hat{H} = \hat{T} + \hat{V} = \hat{H} = -\frac{\hbar^2}{2m}\nabla^2 + V(\vec{r}, t)$  . 98

**homogeneous electron gas** quantum mechanical model of electrons distributed uniformly through space (jellium). 111

**invar effect** invar is an Fe Ni (64/36) alloy that has a low thermal expansion coefficient for a range of temperatures below its curie temperature, and this low expansion is the invar effect. 113

**Ir** Iridium  $Z = 77$ ,  $A_r = 192.22$ . 40

**isostructural transformation** a change from one structure to another. 74

**jellium** quantum mechanical model of electrons distributed uniformly through space (homogeneous electron gas). 107, 111

**Kohn-Sham equations**  $\hat{v}_{KS}(\vec{r}_e, \vec{r}_n) = v_{n-e}(\vec{r}_e, \vec{r}_n) + \int d^3r' \frac{\rho(\vec{r}'_e)}{|\vec{r}'_e - \vec{r}_e|} + v_{xc}[\rho](\vec{r}_e)$ . 107–109

**LD50** Lethal dose in 50 percent of cases. 25

**matrix diagonalization** Take matrix  $A$  and find a diagonal matrix  $D$  such that  $S^{-1}AS = D$ . 113

**mean free path** average (mean) distance a projectile travels before reacting with the target. 21

**Mn** Manganese  $Z = 25$ ,  $A_r = 54.94$ . 3

**Mo** Molybdenum  $Z = 42$ ,  $A_r = 95.95$ . 2

**Neel temperature** temperature at which an antiferromagnetic becomes a paramagnet. 76

**observationally stable** theoretically unstable but yet to be measured as unstable by experiment. 148

**orthorhombic**  $a, b, c, \alpha = \beta = \gamma = 90^\circ$ . 59

**P** Phosphorus  $Z = 15$ ,  $A_r = 30.97$ . 33

**Pauli exclusion principle** half-integer spin particles cannot have the same quantum numbers. 109

**Pd** Palladium  $Z = 46$ ,  $A_r = 106.42$ . 40

**plane wave** a wave of parallel planes of constant frequency that are normal to the direction of propagation. 104

**proof strength** stress at which a material undergoes a permanent extension. 30

**Pt** Platinum  $Z = 78$ ,  $A_r = 195.08$ . 40

**Pu** Plutonium  $Z = 94$ ,  $A_r = 244$ . 3

**Rad** Rad (Rad) this is another measure for absorbed dose and 1 rad is equal to 100 ergs per gram (1 rad = 1.0cGy = 0.01Gy). 22

**Rontgen Equivalent Man** Rontgen Equivalent Man (REM) - equivalent/effective dose - 1 REM equivalent is equal to  $1\text{Rad} \times W_r$  (1cGy  $1\text{Rad} \times W_r$ ) and 1 REM effective is equal to  $1\text{Rad} \times W_r \times W_t$  (1cGy  $1\text{Rad} \times W_r \times W_t$ ). 22

**Ru** Ruthenium  $Z = 44$ ,  $A_r = 101.07$ . 2, 40

**Rutherford** Rutherford (Rd) -  $1.0 \times 10^6$  decays per second. 22

**sensitization** The precipitation of chromium rich carbides at the grain boundaries of stainless steel. 35

**Sievert** Sievert (Sv) - equivalent/effective dose - this is the absorbed dose with radiation weighting (equivalent) and tissue weighting (effective) factors applied  $1\text{Sv}(\text{equivalent}) = 1\text{J/kg} \times W_r$  and  $1\text{Sv}(\text{effective}) = 1\text{J/kg} \times W_r \times W_t$ . 22

**supercritical** Above  $374^\circ\text{C}$  (647K) and 22.1MPa water is no longer a distinct liquid or gas and it becomes supercritical. 11

**Th** Thorium  $Z = 90$ ,  $A_r = 232.04$ . 2, 12

**U** Uranium  $Z = 92$ ,  $A_r = 238.03$ . 2

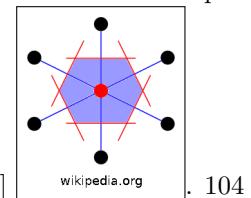
**ultimate tensile strength** maximum stress the material can withstand. 30

**valence electron** outer shell electron that can form bonds. 110

**W** Tungsten  $Z = 74$ ,  $A_r = 183.84$ . 14

**wave function** contains all there is to know about a system and its meaning is given by a probability  $|\Psi|^2 = 1$ . 98

**Wigner-Seitz cell** given a lattice point, the set of all points in space which are closer to that lattice point



that any other lattice point constitute the Wigner-Seitz cell [solidstatebasicswcs]. 104

**Young's modulus** measure of resistance to elastic deformation  $\delta = \frac{\text{stress}}{\text{strain}}$ . 30, 67

**Zr** Zirconium  $Z = 40$ ,  $A_r = 91.22$ . 7

# Bibliography

- [1] D. W. Saxy et al. “Understanding Stress Corrosion Resistant Stainless Steels at the Atomic Scale”. In: (2010).
- [2] World Nuclear. *Advanced Nuclear Power Reactors*. 2020. URL: <https://world-nuclear.org/information-library/nuclear-fuel-cycle/nuclear-power-reactors/advanced-nuclear-power-reactors.aspx>.
- [3] Thomas W. Kerlin and Belle R. Upadhyaya. *Dynamics and Control of Nuclear Reactors*. Vol. 2. 2019.
- [4] H. Khatabil. *Gen-4 Reactors*. 2013. URL: <http://www.gen-4.org/GIF/About/documents/25-Session2-3-Khartabil.pdf>.
- [5] Financial Times. *Cost of new Sizewell C Nuclear Plant*. 2020. URL: <https://www.ft.com/content/77c209f7-6d18-4609-ac3c-77d1b5b82b34>.
- [6] Energy for Generations. *ESB opens 820m Euro Carrington Power Station in Manchester*. 2020. URL: <https://www.esb.ie/tns/press-centre/2017/2017/03/13/esb-opens-820m-carrington-power-station-in-manchester>.
- [7] Our World In Data. *Safest and Cleanest Sources of Energy*. 2020. URL: <https://ourworldindata.org/uploads/2020/10/What-is-the-safest-form-of-energy-2048x1129.png>.
- [8] 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\%5F06F10\%5Flec06a.pdf>.
- [9] Yuchen Dou, Hong Luo, and Jing Zhang. “Elastic Properties of FeCr20Ni8Xn (X = Mo, Nb, Ta, Ti, V, W and Zr) Austenitic Stainless Steels: A First Principles Study”. In: *Metals* 9 (145 2019).
- [10] Philippe F Weck et al. *Mechanical properties of zirconium alloys and zirconium hydrides predicted from density functional perturbation theory*. 2020. URL: <https://www.osti.gov/pages/servlets/purl/1259850>.
- [11] GenIV International Forum. *Introduction to Generation IV Nuclear Energy Systems and the International Forum*. 2013. URL: <https://www.iea.org/topics/world-energy-outlook>.
- [12] Wikipedia. *Carnots theorem (thermodynamics)*. 2020. URL: <https://en.wikipedia.org/wiki/Carnot>.
- [13] S. E. Jensen and E. Nonbol. *Description of the Magnox Type of Gas Cooled Reactor (MAGNOX)*. 1998. URL: <https://inis.iaea.org/collection/NCLCollectionStore/\%5FPublic/30/052/30052480.pdf>.
- [14] Wikipedia. *Combined cycle power plant*. 2020. URL: <https://en.wikipedia.org/wiki/Combined\%5Fcycle\%5Fpower\%5Fplant>.
- [15] World Nuclear. *Uranium Enrichment*. 2020. URL: <https://www.world-nuclear.org/information-library/nuclear-fuel-cycle/conversion-enrichment-and-fabrication/uranium-enrichment.aspx>.

- [16] Los Alamos National Laboratory. *Los Alamos National Laboratory Tour*. 2013. URL: <http://t2.lanl.gov/nis/tour/sch007.html>.
- [17] GenIV International Forum. *Very-High-Temperature Reactor (VHTR)*. 2020. URL: <https://www.gen-4.org/gif/jcms/c%5F9362/vhtr>.
- [18] Sangeeta Deokattey et al. “Hydrogen production using high temperature reactors: an overview”. In: *Advances in Energy Research* 1 (1 2013), pp. 013–033.
- [19] S Heidenreich, M Müller, and P. U. Foscolo. *Advanced Biomass Gasification - New Concepts for Efficiency Increase and Product Flexibility*. 2016.
- [20] 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\%5F06F10\%5Flec06b.pdf>.
- [21] V. Ignatiev and A. Surenkov. *Corrosion phenomena induced by molten salts in Generation IV nuclear reactors*. 2017, p. 166.
- [22] A. Santamarina et al. *The JEFF-3.1.1 Nuclear Data Library*. 2009.
- [23] M. R. Gilbert, J. Marian, and J-Ch. Sublet. “Energy spectra of primary knock-on atoms under neutron irradiation”. In: *Journal of Nuclear Materials* 467 (2015), pp. 121–134.
- [24] J. Fikar and R. Schaublin Ecole. “Molecular dynamics simulation of radiation damage in bcc tungsten”. In: *Journal of Nuclear Materials* (101 2009), pp. 97–101.
- [25] Andrew F. Calder et al. “Computer simulation of cascade damage in a-iron with carbon in solution”. In: *Journal of Nuclear Materials* (382 2008), pp. 91–95.
- [26] N J Carron. *An Introduction to the Passage of Energetic Particles through Matter*. 2007.
- [27] James Byrne. *Neutrons, Nuclei & Matter*. 2011, p. 161.
- [28] G. F. Knoll. *Radiation Detection and Measurement Third Edition*. 1999, pp. 53–54.
- [29] NIST. *X-Ray Mass Attenuation Coefficients*. 2022. URL: <https://physics.nist.gov/PhysRefData/XrayMassCoef/tab3.html>.
- [30] J R Greening. *Fundamentals of Radiation Dosimetry*. 1985.
- [31] UK Statutory Instrument. *The Ionising Radiations Regulations 2017*. 2017. URL: <https://www.legislation.gov.uk/uksi/2017/1075/schedule/3>.
- [32] J. Rodenas and G. Verdu. “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.
- [33] H. Sheng. *EAM Potentials*. 2012. URL: <https://sites.google.com/site/eampotentials>.
- [34] G. S. Was. *Fundamentals of Radiation Materials Science*. 2007.
- [35] R. E. Stoller. “Primary Damage Formation in Irradiated Materials”. In: *Journal of Metals* 48 (12 1996), pp. 23–27.
- [36] Jian wei Su and Dong ping Huang. “Phase equilibria of the Fe–Cr–Ni ternary systems and interfacial-reactions in Fe–Cr alloys with Ni substrate”. In: *Journal of Alloys and Compounds* 457 (2008), pp. 270–278.
- [37] S. Choudhury a et al. “Ab-initio based modelling of diffusion in dilute bcc Fe–Ni and Fe–Cr alloys and implications for radiation induced segregation”. In: *Journal of Materials* 411 (2011), pp. 1–14.
- [38] J. L. Seran and M. Le Flem. *Irradiation-resistant Austenitic Steels as Core Materials for Generation IV Nuclear Reactors*. 2017, pp. 285–328.
- [39] T.R. Allen and J.T. Busby. “Radiation damage concerns for extended light water Reactor service”. In: *Journal of Materials* 61 (2009), pp. 29–34.

- [40] Gary S. Was and Shigeharu Ukai. *Austenitic Stainless Steels*. 2019.
- [41] T.R. Allen et al. “Swelling and radiation-induced segregation in austenitic alloys”. In: *Journal of Nuclear Materials* 342 (2005), pp. 90–100.
- [42] T. R. Allen, J. I. Cole, and E. A. Kenik. *Radiation-Induced Segregation and Void Swelling in 304 Stainless Steel*. 2000. URL: <https://www.osti.gov/servlets/purl/757550>.
- [43] M. Krcmar et al. “Diffusion rates of 3d transition metal solutes in nickel by first-principles calculations”. In: *Acta Materialia* 53 (2005), pp. 2369–2376.
- [44] IAEA Geir Meyer and Egil Stokke. *Description of Sizewell B Nuclear Power Plant*. 2019. URL: <https://inis.iaea.org/collection/NCLCollectionStore/\%5FPublic/29/010/29010110.pdf>.
- [45] G. S. Was. *Fundamentals of Radiation Materials Science*. 2007, p. 183.
- [46] B. Kombaiah et al. “Mechanisms of radiation-induced segregation around He bubbles in a Fe-Cr-Ni crystal”. In: *Journal of Nuclear Materials* 11 (2018).
- [47] W G Johnston, W G Morris, and A M Turkalo. “Proc. Int. Con. on Radiation Effects in Breeder Reactor Structural Materials”. In: *Metall. Soc. AIME* (1977), p. 421.
- [48] A D Marwick. “Segregation in irradiated alloys: The Inverse Kirkendall Effect and the effect of constitution on void swelling”. In: *J. Phys. F: Met. Phys* 8 (1978), pp. 1849–1862.
- [49] P. R. Okamoto and L. E. Rehn. “RADIATION-INDUCED SEGREGATION IN BINARY AND TERNARY ALLOYS”. In: *Journal of Nuclear Materials* 83 (1979), pp. 2–23.
- [50] Jeremy L. Gilber. *Metals - Basic Principles*.
- [51] R. Kirchheim et al. “Compositional Changes of Passive Films Due To Different Transport Rates and Preferential Dissolution”. In: *Corrosion Science* 31 (1990), pp. 573–578.
- [52] R. Kirchheim et al. “The Passivity of Iron-Chromium Alloys”. In: *Corrosion Science* 29 (7 1989), pp. 899–917.
- [53] P. J. Maziasz and M. Le Flem. *Properties of Austenitic Steels for Nuclear Reactor Applications*. Vol. 2. 2012, pp. 303–318.
- [54] Rolled Alloys. *What is alloy sensitization?* 2020. URL: <https://www.rolledalloys.com/dotAsset/50dc766c-08f5-43d9-bbda-5d9a9a2fa6d5.jpg>.
- [55] Speciality Steel Industry of North America. *Intergranular Corrosion*. 2020. URL: <https://www.ssina.com/education/corrosion/intergranular-corrosion/>.
- [56] S. Lyon. *Corrosion of Molybdenum and its Alloys*. 2010. URL: <https://www.researchgate.net/publication/287308921\%5FCorrosion\%5Fof\%5FMolybdenum\%5Fand\%5Fits\%5FAlloys>.
- [57] V. Maurice et al. “Effects of molybdenum on the composition and nanoscale morphology of passivated austenitic stainless steel surfaces”. In: *Faraday Discussions* 180 (2015), pp. 151–170.
- [58] G. S. Was. *Fundamentals of Radiation Materials Science*. 2007, p. 765.
- [59] Edward Kenik, R H Jones, and G E C Bell. “Irradiation-assisted stress corrosion cracking”. In: *Journal of Nuclear Materials* (212-215 1994), pp. 52–59.
- [60] Sophie Le Caer. *Water Radiolysis: Influence of Oxide Surfaces on H<sub>2</sub> Production under Ionizing Radiation*. 2011. URL: <https://www.mdpi.com/2073-4441/3/1/235>.
- [61] R.W. Staehle. *Historical views on stress corrosion cracking of nickel-based alloys: the Coriou effect*. 2016.
- [62] United States Nuclear Regulatory Commission. *AP1000 Design Control Document*. 2020. URL: <https://www.nrc.gov/docs/ML1117/ML11171A447.pdf>.
- [63] P. N. Standring. *The Long Term Storage of Advanced Gas-Cooled Reactor Fuel*. URL: <https://inis.iaea.org/collection/NCLCollectionStore/\%5FPublic/30/040/30040095.pdf>.

- [64] Guy O.H. Whillock et al. "Investigation of thermally sensitised stainless steels as analogues for spent AGR fuel cladding to test a corrosion inhibitor for intergranular stress corrosion cracking". In: *Journal of Nuclear Materials journal* (498 2018), pp. 187–198.
- [65] IAEA. *Status report 81 - Advanced Passive PWR (AP 1000)*. 2020. URL: <https://aris.iaea.org/PDF/AP1000.pdf>.
- [66] Framatome ANP and Inc. *EPR Design Description*. 2020. URL: <https://www.nrc.gov/docs/ML0522/ML052280170.pdf>.
- [67] F. Dalle et al. *Conventional austenitic steels as out-of-core materials for Generation IV nuclear reactors*.
- [68] Pascal V. Grundler and Stefan Ritter. "Nobel Metal Chemical Addition for Mitigation of Stress Corrosion Cracking: Theoretical Insights and Applications". In: *PPChem* (2014), pp. 76–93.
- [69] J. H. Potgieter and A. Van Benekom. "The effect of varying ruthenium content on the corrosion behavior of two cathodically modified superferritic stainless steels". In: *Canadian Metallurgical Quarterly* 34 (2 1994), pp. 143–146.
- [70] IAEA Department of Nuclear Sciences and Applications. *Directory of Cyclotrons used for Radionuclide Production in Member States*. 2006. URL: <http://www-naweb.iaea.org/napc/iachem/cyclotrons/PDF/DCRP.pdf>.
- [71] Diamond. *About Synchrotrons*. 2020. URL: <https://www.diamond.ac.uk/Home/About/FAQs/About-Synchrotrons.html>.
- [72] World Nuclear. *World Nuclear Research Reactors*. 2018. URL: <http://www.world-nuclear.org/information-library/non-power-nuclear-applications/radioisotopes-research/research-reactors.aspx>.
- [73] IAEA. *Nepture Reactor*. 2018. URL: <https://nucleus.iaea.org/RRDB/RR/ReactorSearch.aspx>.
- [74] ORNL. *High Flux Isotope Reactor User Guide*. 2018. URL: <https://neutrons.ornl.gov/sites/default/files/HighFluxIsotopeReactorUserGuide2.0.pdf>.
- [75] ORNL. *High Flux Isotope Reactor Parameters*. 2018. URL: <https://neutrons.ornl.gov/hfir/parameters>.
- [76] ORNL. *High Flux Isotope Reactor (HFIR) USER GUIDE*. 2015. URL: <https://neutrons.ornl.gov/sites/default/files/High%20Flux%20Isotope%20Reacto%20User%20Guide%202.0.pdf>.
- [77] John J. Rush and Ronald L. Cappelletti. *The NIST Center for Neutron Research: Over 40 Years Serving NIST/NBS and the Nation*. 2015. URL: <https://www.ncnr.nist.gov/NCNRHistory%5FRush%5FCappelletti.pdf>.
- [78] ISIS. *How ISIS works - In Depth*. 2020. URL: <https://www.isis.stfc.ac.uk/Pages/How-ISIS-works--in-depth.aspx>.
- [79] ORNL E. B. Iverson. *Neutron Moderation*. 2016. URL: <https://conference.sns.gov/event/56/attachments/64/99/Lecture%5F2a%5F-%5FNeutron%5FModeration%5F-%5FERik%5FIverson.pdf>.
- [80] M. N. H. Comsan. *Spallation Neutron Sources for Science and Technology*. 2011. URL: <https://inis.iaea.org/collection/NCLCollectionStore/\%5FPublic/43/099/43099436.pdf>.
- [81] R. B. Leachman. *Peaceful Uses of Atomic Energy*. Vol. 2. 1956, p. 193.
- [82] Hyperphysics. *Calculation of Coulomb Barrier*. 2022. URL: <http://hyperphysics.phy-astr.gsu.edu/hbase/NucEne/coubar.html>.
- [83] Hyperphysics. *Nuclear Size and Density*. 2022. URL: <http://hyperphysics.phy-astr.gsu.edu/hbase/Nuclear/nucuni.html\#c4>.

- [84] V.V. Zerkin and B. Pritychenko. “The experimental nuclear reaction data (EXFOR): Extended computer database and Web retrieval system”. In: *Nuclear Inst. and Methods in Physics Research, A* 888 (2018), pp. 31–43.
- [85] Shigeo Tanaka and Michiaki Furukawa. “Excitation Functions for (p,n) Reactions with Titanium, Vanadium, Chromium, Iron and Nickel up to Ep”. In: *Journal of the Physical Society of Japan* 14 (10 1959), pp. 1269–1275.
- [86] M. S. Livingston and H. A. Bethe. “Nuclear Physics C. Nuclear Dynamics, Experimental”. In: *Physical Review* 9 (3 1937), pp. 245–398.
- [87] J. Wing and J. R. Huizenga. “(p,n) Cross Sections of V51, Cr52, Cu63, Cu65, Ag107, Ag109, Cd111, Cd112 and La139 from 5 to 10.5MeV”. In: *Physical Review* 128 (1 1962), pp. 280–290.
- [88] E. Gadioli et al. “Excitation Functions of V51, Fe56, 65Cu (p,n) Reactions Between 10 and 45 MeV”. In: *Il Nuovo Cimento* 22 (4 1974), pp. 547–558.
- [89] P E Hodgson. “The Nuclear Optical Model”. In: *Rep. Prog. Phys.* 34 (1971), pp. 765–819.
- [90] A Koning, S Hilaire, and S Goriely. *TALYS 1.95 A Nuclear Reaction Program*. 2019.
- [91] NEA. *Janis Book*. 2020. URL: <https://tendl.web.psi.ch/tendl\%F2019/other/book-protons.pdf>.
- [92] NEA. *Exfor reaction cross section website*. 2020. URL: <https://www-nds.iaea.org/exfor/>.
- [93] J. F. Ziegler, J. P. Biersack, and M. D. Ziegler. *The Stopping and Range of Ions in Matter*. 2015.
- [94] James F. Ziegler, M. D. Ziegler, and J. P. Biersack. “SRIM - The stopping and range of ions in matter”. In: *Nuclear Instruments and Methods in Physics Research Section B* 268 (2010), pp. 1818–1823.
- [95] Roger Rousseau, Vassiliki-Alexandra Glezakou, and Annabella Selloni. “Theoretical insights into the surface physics and chemistry of redox-active oxides”. In: *Nature Reviews Materials* (5 2020), pp. 460–475. URL: <https://www.nature.com/articles/s41578-020-0198-9?proof>.
- [96] F. D. Murnaghan. “The Compressibility of Media Under Extreme Pressures”. In: (1944).
- [97] A. Mahmoud ad A. Erba. *Crystal: Equation of State*. 2019. URL: <http://tutorials.crystalsolutions.eu/tutorial.html?td>.
- [98] F. Birch. “Finite elastic strain of cubic crystals”. In: *Physical Review* 71 (11 1947), pp. 809–824.
- [99] gilgamesh. *Fitting Birch-Murnaghan*. 2013. URL: <http://gilgamesh.cheme.cmu.edu/doc/software/jacapo/appendices/appendix-eos.html>.
- [100] M. J. Mehl, D. J. Singh, and D. A. Papaconstantopoulos. “Properties of ordered intermetallic alloys: first-principles and approximate methods”. In: *Materials Science and Engineering* (1993), pp. 49–57.
- [101] Pascal Vinet et al. “Universal features of the equation of state of solids”. In: *Journal of Physics Condensed Matter* 1 (1989), pp. 1941–1963.
- [102] B Jelinek et al. “Modified embedded atom method potential for Al, Si, Mg, Cu, and Fe alloys”. In: *Physical Review B* 84 (24 2012), pp. 1–18.
- [103] G. Bonny, R. C. Pasianot, and L. Malerba. “Interatomic potentials for alloys: Fitting concentration dependent properties”. In: *Philosophical Magazine* 89 (8 2009), pp. 711–725.
- [104] 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.
- [105] P. Ravindran et al. “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.
- [106] M. J. Mehl, B. M. Klein, and D. A. Papaconstantopoulos. *First Principles Calculations of Elastic Properties of Metals*. 1994.

- [107] B B Karki, G J Ackland, and J Crain. *Elastic instabilities in crystals from ab initio stress-strain relations*. 2013. URL: <http://www.homepages.ed.ac.uk/gja/karki1.pdf>.
- [108] G. Ghosh. “A first-principles study of cementite (Fe<sub>3</sub>C) and its alloyed counterparts: Elastic constants, elastic anisotropies, and isotropic elastic moduli”. In: *AIP Advances* 5 (2015).
- [109] M. E. Fine, L. D. Brown, and H. L. Marcus. “Elastic Constants Versus Melting Temperature in Metals”. In: *Scripta Metallurgica* 18 (1984), pp. 951–956.
- [110] M. W. Finnis and J. E. Sinclair. “A simple empirical N-body potential for transition metals”. In: *Philosophical Magazine A* 50 (1984).
- [111] Murray S. Daw and M. I. Baskes. “Embedded-atom method: Derivation and application to impurities, surfaces and other defects in metals”. In: *Physical Review B* 29 (12 1983).
- [112] Graeme J. Ackland. “Two-band second moment model for transition metals and alloys”. In: *Condensed Materials* (2005).
- [113] P. Olsson et al. “Two-band modelling of alpha prime phase formation in Fe-Cr”. In: *Physical Review B* 72 (2005).
- [114] G. Bonny et al. “Iron chromium potential to model high-chromium ferritic alloys”. In: *Philosophical Magazine* 91 (April 2011), pp. 1724–1746.
- [115] Byeong-Joo Lee, Jae-Hyeok Shim, and M. I. Baskes. “Semiempirical atomic potentials for the fcc metals Cu, Ag, Au, Ni, Pd, Pt, Al, and Pb based on first and second nearest-neighbor modified embedded atom method”. In: *Physical Review B* 68 (2003).
- [116] J. Fikar and R. Schäublin. “Molecular dynamics simulation of radiation damage in bcc tungsten”. In: *Journal of Nuclear Materials* 386-388 (C 2009), pp. 97–101.
- [117] Y. Mishin. “Atomistic modeling of the gamma and gamma prime phases of the Ni-Al system”. In: *Acta Materialia* 52 (2004), pp. 1451–1467.
- [118] M. I. Mendelev et al. “Development of new interatomic potentials appropriate for crystalline and liquid iron”. In: *Philosophical Magazine* 83 (2003), pp. 3977–3994.
- [119] G. Bonny, A. Bakaev, and D. Terentyev. “Interatomic potential to study plastic deformation in tungsten-rhenium alloys”. In: *Journal of Applied Physics* 121 (165107 2017), pp. 1–13.
- [120] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. 2006, p. 449.
- [121] F. Ercolessi and J. B. Adams. “Interatomic Potentials from First-Principles Calculations: the Force-Matching Method”. In: *Europhys. Lett.* 26 (8 1994), pp. 583–588.
- [122] G. Bonny and L. Malerba. “Overview of Interatomic Potentials”. In: *SCK CEB Open Report* (2005).
- [123] Hector Balboa et al. “Damage characterization of (U,Pu)O<sub>2</sub> under irradiation by molecular dynamics simulations”. In: *Journal of Nuclear Materials* 512 (2018), pp. 440–449.
- [124] K. A. Nekrasov and. “Sputtering of material from the surface of PuO<sub>2</sub> crystals by collision cascades impact. A molecular dynamics study”. In: *Journal of Nuclear Materials* 512 (2018), pp. 440–449.
- [125] “Defect production due to displacement cascades in metals as revealed by computer simulation”. In: *Journal of Nuclear Materials* 251 (1997), pp. 1–12.
- [126] K. Trachenko et al. “Modeling high-energy radiation damage in nuclear and fusion applications”. In: *Nuclear Instruments and Methods in Physics Research B* 277 (2012), pp. 6–13.
- [127] I.T.Todorov and W. Smith. *The DL\_POLY\_4 User Manual*. 2016.
- [128] *LAMMPS - Partitioning*. URL: [https://docs.lammps.org/Developer\\_par\\_part.html](https://docs.lammps.org/Developer_par_part.html).
- [129] Wikipedia. *Virial Stress*. 2016. URL: <https://en.wikipedia.org/wiki/Virial\%5Fstress>.

- [130] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi. “Optimization by Simulated Annealing”. In: *Science* 220 (4598 1983), pp. 671–680.
- [131] David J. Wales and Jonathan P. K. Doye. “Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms”. In: *J. Phys. Chem. A* 1997 101 (1997), pp. 5111–5116.
- [132] *SciPy Basin Hopping Algorithm*.
- [133] Jonathan Richard Shewchuk. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain Edition 1.25*. 1994. URL: <https://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf>.
- [134] Derek J Hepburn and Graeme J Ackland. “Metallic-covalent interatomic potential for carbon in iron”. In: *Physical Review B* 78 (165115 2008), pp. 1–6.
- [135] P. Brommer and F. Gähler. “Potfit: effective potentials from ab initio data”. In: *Modelling Simul. Mater. Sci. Eng.* 15 (2007), pp. 295–304.
- [136] H. W. Sheng et al. “Highly Optimized Embedded-Atom-Method Potentials for Fourteen FCC Metals”. In: *Europhys. Lett.* 26 (8 1994), pp. 583–588.
- [137] D. Connetable and O. Thomas. “First-principles study of the structural, electronic, vibrational and elastic properties of orthorhombic NiSi”. In: *Physical Review B* 79 (2009).
- [138] Materials. *Microstructure and Mechanical Properties of*. 2019. URL: <https://www.mdpi.com/1996-1944/12/2/290/pdf\#:~:text>.
- [139] Steven H. Simon. *Solid State Basics*. 2013.
- [140] MIT. *Physics for Solid State Applications*. 2004. URL: <http://web.mit.edu/6.730/www/ST04/Lectures/Lecture7.pdf>.
- [141] Harry Jones. *The Theory of Brillouin Zones and Electronic States in Crystals*. 1960, p. 48.
- [142] P. Hohenberg and W. Kohn. “Inhomogeneous Electron Gas”. In: *Physical Review* 136 (1964).
- [143] Robert van Leeuwen. *Introduction to density-functional theory*. 2020. URL: <http://users.jyu.fi/~roleeuwe/DFTLectureNotes.pdf>.
- [144] W. Kohn and L. J. Sham. “Self-Consistent Equations Including Exchange and Correlation Effects”. In: *Physical Review* 140 (1965).
- [145] SISSA. *ABC of DFT*. 2020. URL: <https://people.sissa.it/~degironc/ES/lectures/ABCofDFT.pdf>.
- [146] J. F. Annett. “Efficiency of algorithms for Kohn-Sham density functional theory”. In: *Computational Materials Science* 4 (1995), pp. 23–42.
- [147] G. Kresse and J. Furthmüller. “Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set”. In: *Physical Review B* 54 (16 1996).
- [148] D. Raczkowski, A. Canning, and L. W. Wang. “Thomas-Fermi charge mixing for obtaining self-consistency in density functional calculations”. In: *Physical Review B* 64 (2001).
- [149] D. D. Johnson. “Modified Broydens method for accelerating convergence in self-consistent calculations”. In: *Physical Review B* 38 (18 1988).
- [150] P. Ziesche, S. Jurth, and J. P. Perdew. “Density functionals from LDA to GGA”. In: *Computational Materials Science* 11 11 (1998), pp. 122–127.
- [151] M. C. Payne et al. “Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients”. In: *REVIEWS OF MODERN PHYSICS* 64 (1992).
- [152] Annabella Selloni. *The plane wave-pseudopotential method (basic aspects)*. 2009. URL: <https://th.fhi-berlin.mpg.de/th/Meetings/DFT-workshop-Berlin2009/Talks/OnlinePublication/0624-2\%5F20090623-1\%5FSelloni\%5FFINAL\%5F2perA4\%5F-\%5FSelloni-Berlin2.pdf>.

- [153] V.P. Gupta. *PRINCIPLES AND APPLICATIONS OF QUANTUM CHEMISTRY - Density Functional Theory (DFT) and Time Dependent DFT (TDDFT)*. 2016, p. 169.
- [154] John P Perdew, Kieron Burke, and Matthias Ernzerhof. “Generalized Gradient Approximation Made Simple”. In: *Physical Review Letters* 77 (18 1996), pp. 3865–3868.
- [155] John P Perdew. “Generalized Gradient Approximations for exchange and correlation: A look backward and forward”. In: *Physica B* 172 (1991), pp. 1–6.
- [156] M R Benam, N Abdoshahi, and M Majidiyan Sarmazdeh. “Ab initio study of the effect of pressure on the structural and electronic properties of cubic LaAlO<sub>3</sub> by density function theory using GGA, LDA and PBESOL exchange correlation potentials”. In: *Physica B* 446 (2014), pp. 32–38.
- [157] periodictable.com. *Sodium: periodictable.com*. 2020. URL: <https://periodictable.com/Elements/011/data.html>.
- [158] periodictable.com. *Aluminium: periodictable.com*. 2020. URL: <https://periodictable.com/Elements/013/data.html>.
- [159] P Giannozzi et al. “Quantum Espresso”. In: *J Phys Condens Matter* 29 (2017).
- [160] University of Illinois. *Density functional theory calculation of elastic constants*. 2020. URL: <https://courses.physics.illinois.edu/mse404ela/sp2018/Project2-DFT.html>.
- [161] Alessandro Genova and Michele Pavanello. “Exploiting the Locality of Periodic Subsystem Density-Functional Theory: Efficient Sampling of the Brillouin Zone”. In: *J. Phys.: Condens. Matter* 27 (2015), pp. 495–501. URL: <https://iopscience.iop.org/article/10.1088/0953-8984/27/49/495501/ampdf>.
- [162] N. Marzari. *Ab-initio Molecular Dynamics for Metallic Systems*. 1996, p. 77.
- [163] Hyperphysics. *Electron band gap diagram for insulators, semiconductors and conductors*. 2020. URL: <http://hyperphysics.phy-astr.gsu.edu/hbase/Solids/band.html>.
- [164] University of Cambridge. *Dissemination of IT for the Promotion of Materials Science*. 2020. URL: <https://www.doitpoms.ac.uk/tlplib/semiconductors/fermi.php>.
- [165] M. Methfessel and A. T. Paxton. “High-precision sampling for Brillouin-zone integration in metals”. In: *Physical Review B* 40 (6 1989), pp. 3616–3621.
- [166] N. Marzari. *Ab-initio Molecular Dynamics for Metallic Systems*. 1996, p. 136.
- [167] Steven H. Simon. *Solid State Basics*. 2013, p. 213.
- [168] H. Stehfest. “Algorithm 368 Numerical Inversion of Laplace Transform”. In: *Communications of the ACM* 13 (1970), pp. 47–49.
- [169] J. Ziegler. *The Stopping and Range of Ions in Matter*. 2019. URL: <http://www.srim.org/SRIM/SRIMINTRO.htm>.
- [170] A.J. Koning and D. Rochman. “TENDL: Comprehensive Nuclear Data Library with Covariances”. In: (2010).
- [171] A.J. Koning, D. Rochman, and J. Sublet. “TENDL: Comprehensive Nuclear Data Library with Covariances”. In: (2017).
- [172] A.J. Koning et al. “TENDL: Complete Nuclear Data Library for Innovative Nuclear Science and Technology”. In: *Nuclear Data Sheets* 155 (2019).
- [173] M.C.Lagunas-Solar and J.A.Jungerman. “Cyclotron production of carrier-free cobalt-55.a new positron-emitting label for bleomycin”. In: *Applied Radiation and Isotopes* 30 (1979), p. 25.
- [174] D. Rochman and A. J. Koning. “Modern Nuclear Data Evaluation With The TALYS Code System”. In: *Nuclear Data Sheets* 113 (2012).

- [175] J. Hewett. *Use of the University of Birmingham's Scanditronix MC40 Cyclotron for Materials Irradiation Damage Studies*. 2011.
- [176] E. Lamere et al. "Proton-induced reactions on molybdenum". In: *Physical Review, Part C, Nuclear Physics* 100 (2019), p. 034614.
- [177] D.S.Flynn, R.L.Hershberger, and F.Gabbard. "Sub-Coulomb proton absorption for isotopes of zirconium and molybdenum". In: *Physical Review, Part C, Nuclear Physics* 20 (5 1979), p. 1700.
- [178] J.J.Hogan. "Mo-96(P,XN)Reaction from 10 to 80 MeV". In: *Physical Review, Part C, Nuclear Physics* 6 (1972), p. 810.
- [179] V.N.Levkovski. "Mo-96(P,XN)Reaction from 10 to 80 MeV". In: *Soviet Atomic Energy* 69 (1990), p. 773.
- [180] webelements.com. *Webelements: Aluminium*. 2020. URL: <https://www.webelements.com/aluminium>.
- [181] webelements.com. *Webelements: Iron*. 2020. URL: <https://www.webelements.com/iron>.
- [182] webelements.com. *Webelements: Nickel*. 2020. URL: <https://www.webelements.com/nickel>.
- [183] webelements.com. *Webelements: Ruthenium*. 2020. URL: <https://www.webelements.com/ruthenium>.
- [184] webelements.com. *Webelements: Palladium*. 2020. URL: <https://www.webelements.com/palladium>.
- [185] N. Marzari et al. "Thermal Contraction and Disordering of the Al(110) Surface Nicola". In: *Condensed Matter* (110 2008), pp. 4–7.
- [186] A. L. Okana-Lomanga et al. "Adsorption of Br<sub>2</sub> molecule on the Fe W(110) substrate - Energetics, electronic and magnetic properties". In: *Computational Condensed Matter* 23 (2020).
- [187] A. E. Rezaee and M. Almasi-Kashi. "The influence of point defects on Na diffusion in black phosphorene - first principles study". In: *Journal Pre-proof* () .
- [188] N. Acharya and S. P. Sanyal. "Structural phase transition, electronic and superconducting properties of ScBi and YBi". In: *Solid State Communications* 266 (2017), pp. 39–45.
- [189] A. Choudhary, L. Malakkal, and R. K. Siripurapu. "First principles calculations of hydrogen storage on Cu and Pd-decorated graphene". In: *International Journal of Hydrogen Energy* 41 (2016), pp. 17652–17656.
- [190] O. E. Osafie, P. O. Adebambo, and G. A. Adebayo. "Elastic constants and observed ferromagnetism in inverse Heusler alloy Ti<sub>2</sub>CoAs using kjpaw pseudopotentials - A first-principles approach". In: *Journal of Alloys and Compounds* 722 (2017), pp. 207–211.
- [191] Quantum Espresso. *QE Forum*. 2022. URL: <https://lists.quantum-espresso.org/>.
- [192] *Materials Square Single Atom Calculations*.
- [193] Andrea Fortini, Mikhail I Mendelev, and Sergey Buldyrev et al. "Asperity contacts at the nanoscale: Comparison of Ru and Au". In: *J. Appl. Phys.* 104 (074320 2008), pp. 1–9.

# Chapter 13

## Acknowledgements

Thank you to Brian, Mark and Alessandro for your supervision and support, to my wife Rachel, son Zakk, parents and my family for encouragement. The various forums in the scientific community, in particular useful discussions with Dr. Todorov of Daresbury and members of the Quantum Espresso forum, have been helpful throughout. Two teachers in particular from my school days, Mr. George and Mr. Rockett for their encouragement in the sciences, especially Physics.

The computations described in this work were performed using the University of Birmingham's BlueBEAR HPC service, which provides a High Performance Computing service to the University's research community. See <http://www.birmingham.ac.uk/bear> for more details.

# **Appendices**

# Appendix A

## Useful Laplace Transforms

### A.1 Table of Transforms

$f(t) = \mathcal{L}^{-1}\{F(s)\}$	$F(s) = \mathcal{L}\{f(t)\}$
$f(t)$	$F(s)$
$f'(t)$	$sF(s) - f(0)$
$a$	$\frac{1}{s+a}$
$\exp(-at)$	$\frac{1}{s+a}$
$a \exp(-bt)$	$\frac{a}{s+b}$
$\frac{1}{a} [1 - \exp(-at)]$	$\frac{1}{s(s+a)}$
$\frac{a}{b} [1 - \exp(-bt)]$	$\frac{a}{s(s+b)}$

Table A.1: Useful Laplace transforms

### A.2 Decay with a Source Term

Set up the equation:

$$\frac{dN(t)}{dt} = \omega - \lambda N(t) \quad (\text{A.1})$$

Transform:

$$\begin{aligned} \mathcal{L} \left\{ \frac{dN(t)}{dt} \right\} &= sF(s) - N_0 \\ \mathcal{L} \{\omega - \lambda N(t)\} &= \frac{\omega}{s} - \lambda N(s) \end{aligned} \quad (\text{A.2})$$

Rearrange to make  $N(s)$  the subject:

$$\begin{aligned}
 sN(s) - N_0 &= \frac{\omega}{s} - \lambda N(s) \\
 (s + \lambda)N(s) &= \frac{\omega}{s} + N_0 \\
 N(s) &= \frac{\omega}{s(s + \lambda)} + \frac{N_0}{(s + \lambda)}
 \end{aligned} \tag{A.3}$$

Inverse transform:

$$\begin{aligned}
 \mathcal{L}^{-1}\{N(s)\} &= N(t) \\
 \mathcal{L}^{-1}\left\{\frac{\omega}{s(s + \lambda)}\right\} &= \frac{\omega}{\lambda}(1 - \exp(-\lambda t)) \\
 \mathcal{L}^{-1}\left\{\frac{N_0}{s + \lambda}\right\} &= N_0 \exp(-\lambda t)
 \end{aligned} \tag{A.4}$$

Simplify:

$$\begin{aligned}
 N(t) &= \frac{\omega}{\lambda}(1 - \exp(-\lambda t)) + N_0 \exp(-\lambda t) \\
 N(t) &= \frac{\omega}{\lambda} + \left(N_0 - \frac{\omega}{\lambda}\right) \exp(-\lambda t)
 \end{aligned} \tag{A.5}$$

## Appendix B

# Activity Equation in Python: decay class

### B.1 Full Source Code

The full source code is available to download from github.

[https://github.com/BenPalmer1983/atomic\\_dictionaries](https://github.com/BenPalmer1983/atomic_dictionaries)

<https://github.com/BenPalmer1983/decay>

### B.2 Highlighted Code

The code below is taken from the decay class and it is specifically responsible for calculating the amount of isotopes in the decay chain at some time t.

Listing B.1: Decay python3 class for calculating the amount of an isotope in a decay chain at time t

```
1 import os
2 import numpy
3 from pz import pz
4 from isotopes import isotopes
5 import matplotlib.pyplot as plt
6 import copy
7 import hashlib
8
9 class decay:
10
11     path_isotopes = "../data/isotopes.pz"
12     loaded = False
13
14     @staticmethod
15     def set(path_isotopes):
16         decay.path_isotopes = path_isotopes
17         isotopes.set(path_isotopes)
18         decay.load()
19
20     @staticmethod
21     def load():
22         if(decay.loaded == False):
23             decay.loaded = True
```

```

24
25
26     @staticmethod
27     def chain_isotopes(key, out=[]):
28         if(not isotopes.is_valid(key)):
29             return out
30         if(key not in out):
31             out.append(key)
32         else:
33             return out
34         if(isotopes.is_stable(key)):
35             return out
36         else:
37             dm = isotopes.get_decay_modes(key)
38             for k in dm.keys():
39                 decay.chain_isotopes(k, out)
40             return out
41
42
43     @staticmethod
44     def make_chain(key, l=0, out=[], bf=0.0):
45         if(not isotopes.is_valid(key)):
46             return out
47         if(l == 0):
48             decay.chains_store = []
49             out.append([l, key, bf])
50         if(isotopes.is_stable(key)):
51             if(len(out) > 1):
52                 for i in range(len(out)-1,1,-1):
53                     if(out[i-1][0]>=out[i][0]):
54                         out.pop(out[i-1][0])
55             return decay.chains_store.append(copy.deepcopy(out))
56         else:
57             l = l + 1
58             dms = isotopes.get_decay_modes(key)
59             for k in dms.keys():
60                 bf = dms[k]['branching_factor']
61                 decay.make_chain(k, l, out, bf)
62             return out
63
64     @staticmethod
65     def calculate(parent, time, i_data_in, log=None, custom_chain=None):
66
67         if(log != None):
68             log_dir = decay.get_file_dir(log)
69             print(log_dir)
70             decay.make_dir(log_dir)
71
72         decay.results = {
73             'tally': {},
74             'unique': None,
75             'chains': None,
76         }
77         decay.chains_store = None
78
79         if(custom_chain == None):
80             decay.chains_store = []
81             decay.make_chain(parent, 0, [])
82             decay.results['chains'] = []
83             cn = 0
84             for chain in decay.chains_store:

```

```

85     decay.results['chains'].append([])
86     for iso in chain:
87         k = iso[1]
88         bf = iso[2]
89         i_data = isotopes.get(k)
90         half_life = None
91         decay_constant = None
92         n0 = 0.0
93         w = 0.0
94         if(not i_data['stable']):
95             half_life = i_data['half_life']
96             decay_constant = i_data['decay_constant']
97             if(k in i_data_in.keys()):
98                 n0 = i_data_in[k]['n0']
99                 w = i_data_in[k]['w']
100            d = {
101                'isotope_key': k,
102                'bf': bf,
103                'w': w,
104                'n0': n0,
105                'nend': 0,
106                'half_life': half_life,
107                'decay_constant': decay_constant,
108            }
109            decay.results['chains'][cn].append(d)
110            cn = cn + 1
111        else:
112            decay.results['chains'] = custom_chain
113
114
115 # Find unique and make tally
116 decay.results['unique'] = []
117 for chain in decay.results['chains']:
118     for iso in chain:
119         k = iso['isotope_key']
120         if(k not in decay.results['tally'].keys()):
121             decay.results['unique'].append(k)
122             # Use provided data
123             if(iso['half_life'] == None):
124                 stable = True
125                 half_life = None
126                 decay_constant = 0.0
127             else:
128                 stable = False
129                 half_life = iso['half_life']
130                 decay_constant = numpy.log(2) / iso['half_life']
131                 w = iso['w']
132                 n0 = iso['n0']
133                 # Get proton/neutron etc from isotopes database
134                 i_data = isotopes.get(k)
135                 if(i_data is None):
136                     decay.results['tally'][k] = {
137                         'printable': 'Custom',
138                         'element': 'ZZ',
139                         'protons': 999,
140                         'nucleons': 999,
141                         'metastable': 9,
142                         'stable': stable,
143                         'half_life': half_life,
144                         'decay_constant': decay_constant,
145                         'w': w,

```

```

146             'n0': n0,
147             'nend': 0.0,
148         }
149     else:
150         decay.results['tally'][k] = {
151             'printable': decay.pad(isotopes.get_printable_name(k), 12),
152             'element': i_data['element'],
153             'protons': i_data['protons'],
154             'nucleons': i_data['nucleons'],
155             'metastable': i_data['metastable'],
156             'stable': stable,
157             'half_life': half_life,
158             'decay_constant': decay_constant,
159             'w': w,
160             'n0': n0,
161             'nend': 0.0,
162         }
163
164
165 decay.results['chains_individual'] = []
166 for cn in range(len(decay.results['chains'])):
167     for n in range(len(decay.results['chains'][cn])):
168         nc = []
169         if(decay.results['chains'][cn][n]['n0']>0.0 or decay.results['chains'][cn][n]['w']>0.0):
170             for j in range(n, len(decay.results['chains'][cn])):
171                 iso = copy.deepcopy(decay.results['chains'][cn][j])
172                 if(j>n):
173                     iso['n0'] = 0.0
174                     iso['w'] = 0.0
175                 nc.append(iso)
176         if(len(nc)>0):
177             decay.results['chains_individual'].append(nc)
178
179 for cn in range(len(decay.results['chains_individual'])):
180     chain = decay.results['chains_individual'][cn]
181     n0 = numpy.zeros((len(chain),),)
182     w = numpy.zeros((len(chain),),)
183     l = numpy.zeros((len(chain),),)
184     b = numpy.zeros((len(chain)-1,),)
185     for n in range(len(decay.results['chains_individual'][cn])):
186         k = decay.results['chains_individual'][cn][n]['isotope_key']
187         n0[n] = decay.results['chains_individual'][cn][n]['n0']
188         w[n] = decay.results['chains_individual'][cn][n]['w']
189         l[n] = decay.results['tally'][k]['decay_constant']
190         if(n>0):
191             b[n-1] = decay.results['chains_individual'][cn][n]['bf']
192         # Some observationally stable will still have a decay constant, so set to -1.0
193         if(isotopes.is_stable(k)):
194             l[n] = -1.0
195     nt = decay.calculate_activity(time, l, b, w, n0)
196     for n in range(len(decay.results['chains_individual'][cn])):
197         decay.results['chains_individual'][cn][n]['nend'] = nt[n]
198
199 set = []
200 for cn in range(len(decay.results['chains_individual'])):
201     ckey = ''
202     for n in range(len(decay.results['chains_individual'][cn])):
203         k = decay.results['chains_individual'][cn][n]['isotope_key']
204         ckey = ckey + str(decay.results['chains_individual'][cn][n]['isotope_key']) + "NO:" + str(decay.results['chains_individual'][cn][n]['n0']) + "W:" + str(decay.results['chains_individual'][cn][n]['w'])
205     ckeyh = hashlib.md5(ckey.encode())

```

```

206     ckeyh = ckeyh.hexdigest()
207     if(ckeyh not in set):
208         decay.results['tally'][k]['nend'] = decay.results['tally'][k]['nend'] + decay.results['chains_individual']
209             ][cn][n]['nend']
210         set.append(ckeyh)
211
212     # Log
213     if(log != None):
214         width = 140
215         fh = open(log, 'w')
216         fh.write("Unique Isotopes\n")
217         fh.write(decay.hr(width) + "\n")
218         for k in decay.results['unique']:
219             line = decay.results['tally'][k]['printable']
220             fh.write(line + "\n")
221             fh.write("\n")
222             fh.write("\n")
223             fh.write("Decay Chains\n")
224             fh.write(decay.hr(width) + "\n")
225             fh.write("\n")
226             fh.write("\n")
227             for cn in range(len(decay.results['chains'])):
228                 chain = decay.results['chains'][cn]
229
230                 fh.write(decay.pad(cn+1,6))
231                 for n in range(len(chain)):
232                     iso = chain[n]
233                     k = iso['isotope_key']
234                     if(n>0):
235                         bf = "{0:3e}".format(iso['bf'])
236                         fh.write(" --[" + str(bf) + "]--> ")
237                         fh.write(decay.pad(decay.results['tally'][k]['printable'], 12))
238                         fh.write("\n")
239                         fh.write(decay.pad("T1/2",6))
240                         for n in range(len(chain)):
241                             if(n>0):
242                                 fh.write(decay.pad("",17))
243                                 if(decay.results['chains'][cn][n]['half_life'] == None):
244                                     fh.write(decay.pad("[Stable]",12))
245                                 else:
246                                     fh.write(decay.pad("[" + str("{0:8e}".format(decay.results['chains'][cn][n]['half_life'])) + "]",12))
247                                     fh.write("\n")
248                                     fh.write(decay.pad("L",6))
249                                     for n in range(len(chain)):
250                                         if(n>0):
251                                             fh.write(decay.pad("",17))
252                                             if(decay.results['chains'][cn][n]['decay_constant'] == None):
253                                                 fh.write(decay.pad("[Stable]",12))
254                                             else:
255                                                 fh.write(decay.pad("[" + str("{0:8e}".format(decay.results['chains'][cn][n]['decay_constant'])) + "]",12))
256                                                 fh.write("\n")
257                                                 fh.write("\n")
258                                                 fh.write("\n")
259
260
261         fh.write("Amounts\n")
262         fh.write(decay.hr(width) + "\n")
263         fh.write("\n")

```

```

265     fh.write(decay.hr(width) + "\n")
266     line = decay.pad("Isotope", 12)
267     line = line + decay.pad("T(1/2)", 18)
268     line = line + decay.pad("Decay Constant", 18)
269     line = line + decay.pad("W", 18)
270     line = line + decay.pad("N(t=0)", 18)
271     line = line + decay.pad("N(t=" + str(time) + ")", 18)
272     line = line + decay.pad("A(t=0)", 18)
273     line = line + decay.pad("A(t=" + str(time) + ")", 18)
274     fh.write(line + "\n")
275     fh.write(decay.hr(width) + "\n")
276
277
278     for k in decay.results['tally'].keys():
279         line = decay.pad(decay.results['tally'][k]['printable'], 12)
280         if(decay.results['tally'][k]['half_life'] is None):
281             line = line + decay.pad("Stable", 18)
282             line = line + decay.pad("Stable", 18)
283         else:
284             line = line + decay.pad(str("{0:16e}").format(decay.results['tally'][k]['half_life'])).strip(), 18)
285             line = line + decay.pad(str("{0:16e}").format(decay.results['tally'][k]['decay_constant'])).strip(), 18)
286             line = line + decay.pad(str("{0:16e}").format(decay.results['tally'][k]['w'])).strip(), 18)
287             line = line + decay.pad(str("{0:16e}").format(decay.results['tally'][k]['n0'])).strip(), 18)
288             line = line + decay.pad(str("{0:16e}").format(decay.results['tally'][k]['nend'])).strip(), 18)
289         if(decay.results['tally'][k]['half_life'] is not None):
290             line = line + decay.pad(str("{0:16e}").format(decay.results['tally'][k]['decay_constant'] * decay.results
291                 ['tally'][k]['n0'])).strip(), 18)
292             line = line + decay.pad(str("{0:16e}").format(decay.results['tally'][k]['decay_constant'] * decay.results
293                 ['tally'][k]['nend'])).strip(), 18)
294         fh.write(line + "\n")
295         fh.write(decay.hr(width) + "\n")
296         fh.write("\n")
297         fh.write("\n")
298
299     return decay.results
300
301 ######
302 # DECAY EQUATIONS
303 #####
304
305 @staticmethod
306 def calculate_activity(t, lam, b, w, n0):
307     nt = numpy.zeros((len(n0),))
308     for m in range(0,len(n0)):
309         if(lam[m] <= 0):
310             nt[m] = decay.calc_stable(t, m, lam, b, w, n0)
311         else:
312             nt[m] = decay.calc_unstable(t, m, lam, b, w, n0)
313     return nt
314
315 @staticmethod
316 def calc_unstable(t, m, lam, b, w, n0):
317     y = 0.0
318     k = 0
319     while(k<=m):
320         y = y + decay.r(k, m, lam, b) * (decay.f_unstable(t, k, m, lam) * n0[k] + decay.g_unstable(t, k, m, lam) * w
321             [k])
322         k = k + 1
323     return y

```

```

323
324     @staticmethod
325     def f_unstable(t, k, m, lam):
326         s = 0.0
327         i = k
328         while(i<=m):
329             p = decay.lprod(lam, k, m, 3, i)
330             s = s + numpy.exp(-lam[i] * t) * (1.0 / p)
331             i = i + 1
332             s = s * (-1)**(m-k)
333         return s
334
335
336     @staticmethod
337     def g_unstable(t, k, m, lam):
338         # Term a
339         p = decay.lprod(lam, k, m, 1)
340         a = 1.0 / p
341         # Term b
342         s = 0.0
343         i = k
344         while(i<=m):
345             p = decay.lprod(lam, k, m, 3, i)
346             s = s + (1 / lam[i]) * numpy.exp(-lam[i] * t) * (1.0 / p)
347             i = i + 1
348             s = s * (-1)**(m-k+1)
349         return a + s
350
351     @staticmethod
352     def calc_stable(t, m, lam, b, w, n0):
353         y = n0[m] + w[m] * t
354         k = 0
355         while(k<=m-1):
356             y = y + decay.r(k, m, lam, b) * (decay.f_stable(t, k, m, lam) * n0[k] + decay.g_stable(t, k, m, lam) * w[k])
357             k = k + 1
358         return y
359
360     @staticmethod
361     def f_stable(t, k, m, lam):
362         mm = m - 1
363         p = decay.lprod(lam, k, mm, 1)
364         a = 1.0 / p
365
366         s = 0.0
367         i = k
368         while(i<=mm):
369             p = decay.lprod(lam, k, mm, 3, i)
370             s = s + (1 / lam[i]) * numpy.exp(-lam[i] * t) * (1.0 / p)
371             i = i + 1
372             s = s * (-1)**(mm-k+1)
373         return a + s
374
375     @staticmethod
376     def g_stable(t, k, m, lam):
377         mm = m - 1
378
379         p = decay.lprod(lam, k, mm, 1)
380         a = (t/p)
381
382         s = 0.0
383         i = k

```

```

384     while(i<=mm):
385         p = decay.lprod(lam, k, mm, 4, i)
386         s = s + p
387         i = i + 1
388     q = decay.lprod(lam, k, mm, 2)
389     b = (-s / q)
390
391     c = 0.0
392     i = k
393     while(i<= mm):
394         p = lam[i]**2 * decay.lprod(lam, k, mm, 3, i)
395         c = c + numpy.exp(-1.0 * lam[i] * t) / p
396         i = i + 1
397     c = c * (-1.0)**(mm - k)
398     nd = a + b + c
399
400     return nd
401
402
403 @staticmethod
404 def r(k, m, lam, b):
405     if(k == m):
406         return 1
407     else:
408         p = 1.0
409         i = k
410         while(i<=(m-1)):
411             p = p * b[i] * lam[i]
412             i = i + 1
413         return p
414
415 @staticmethod
416 def lprod(lam, k, m, t=1, i=None):
417     # PROD k,m lam[j]
418     if(t == 1):
419         p = 1.0
420         j = k
421         while(j<=m):
422             p = p * lam[j]
423             j = j + 1
424         return p
425     # PROD k,m lam[j]**2
426     elif(t == 2):
427         p = 1.0
428         j = k
429         while(j<=m):
430             p = p * lam[j]**2
431             j = j + 1
432         return p
433     # PROD k,m, i!=j (lam[i]-lam[j])
434     elif(t == 3):
435         p = 1.0
436         j = k
437         while(j<=m):
438             if(i != j):
439                 p = p * (lam[i] - lam[j])
440                 j = j + 1
441         return p
442     # PROD k,m, i!=j lam[j]
443     elif(t == 4):
444         p = 1.0

```

```

445     j = k
446     while(j<=m):
447         if(i != j):
448             p = p * lam[j]
449         j = j + 1
450     return p
451
452 @staticmethod
453 def pad(inp, l=16):
454     inp = str(inp)
455     while(len(inp)<l):
456         inp = inp + " "
457     return inp
458
459 @staticmethod
460 def hr(l=16):
461     inp = ""
462     while(len(inp)<l):
463         inp = inp + "="
464     return inp
465
466 @staticmethod
467 def get_file_dir(file_path):
468     file_path = file_path.strip()
469     if(file_path[0] != "/"):
470         root = os.getcwd()
471         file_path = root + "/" + file_path
472     file_path = file_path.split("/")
473     path = ""
474     for i in range(1,len(file_path) - 1):
475         path = path + "/" + file_path[i]
476     return path
477
478 @staticmethod
479 def make_dir(dir):
480     dirs = dir.split("/")
481     try:
482         dir = ''
483         for i in range(len(dirs)):
484             dir = dir + dirs[i]
485             if(not os.path.exists(dir) and dir.strip() != ''):
486                 os.mkdir(dir)
487             dir = dir + '/'
488     return True
489 except:
490     return False
491
492
493 @staticmethod
494 def test():
495     print("Test Decay")
496
497     # Load isotopes dictionary
498     decay.set("../data/isotopes.pz")
499
500     idata = {}
501     parent = 84216
502     time = 10
503     idata[84216] = {'w': 0.20, 'n0': 100.0}
504     idata[82212] = {'w': 0.0, 'n0': 5.0}
505     idata[83212] = {'w': 0.07, 'n0': 15.0}

```

```
506     idata[81208] = {'w': 0.005, 'n0': 0.0}
507     # 84Po212 0 0 (default)
508     idata[82208] = {'w': 0.01, 'n0': 300.0}
509     decay.calculate(parent, time, idata, "testing/log_84216_new.txt")
510
511
512 def main():
513     decay.test()
514
515 if __name__ == "__main__":
516     main()
```

---

## B.3 Activity Equation Testing

### B.3.1 Numeric Fortran Code

A simple Fortran code was created for each decay path used to test the activity equation and computer code. The version of the code used to calculate the amount of each isotope for the Po-216 decay chain is given in fig. ??, and this code was modified for each decay chain tested.

Listing B.2: Bash file to compile Fortran test code

---

```

1 #!/bin/bash
2 mpif90 -g \
3 -fopenmp \
4 -fbounds-check \
5 -mtune=native \
6 kinds.f90 \
7 main.f90 \
8 -o test.x \
9 && ./test.x
```

---

Listing B.3: Fortran code to numerically estimate the amount of each isotope in the Polonium-216 decay chain

---

```

1 PROGRAM main_test
2 ! University of Birmingham
3 ! Ben Palmer
4 Use kinds
5 Use MPI
6
7 IMPLICIT NONE
8
9 CALL main()
10
11 CONTAINS
12
13 ! Subroutines
14
15
16 SUBROUTINE main()
17 !#####
18 ! PRIVATE VARIABLES
19 INTEGER(kind=StandardInteger) :: i, j, k
20 INTEGER(kind=StandardInteger) :: steps, chn
21 REAL(kind=DoubleReal) :: w(1:20)
22 REAL(kind=DoubleReal) :: t(1:20)
23 REAL(kind=DoubleReal) :: l(1:20)
24 REAL(kind=DoubleReal) :: n0(1:20)
25 REAL(kind=DoubleReal) :: n(1:20)
26 REAL(kind=DoubleReal) :: source(1:20)
27 REAL(kind=DoubleReal) :: loss(1:20)
28 REAL(kind=DoubleReal) :: t_end, t_step
29 !#####
30
31 print *, "Decay Test - Po216"
32
33 w(:) = 0.0D0
34 l(:) = 0.0D0
35 t(:) = 0.0D0
36 n0(:) = 0.0D0
37 source(:) = 0.0D0
38 loss(:) = 0.0D0
39 chn = 6
```

---

```

40
41 w(:) = 0.0D0
42 w(1) = 0.2D0
43 w(3) = 0.07D0
44 w(4) = 0.005D0
45 w(6) = 0.01D0
46
47 l(1) = 4.683427D+00 ! 84Po216
48 l(2) = 1.809595D-05 ! 82Pb212
49 l(3) = 1.908235D-04 ! 83Bi212
50 l(4) = 3.777781D-03 ! 81Tl208
51 l(5) = 2.310491D+06 ! 84Po212
52 ! 82Pb208
53
54 n0(:) = 0.0D0
55 n0(1) = 1.0D2
56 n0(2) = 5.0D0
57 n0(3) = 1.5D1
58 n0(6) = 3.0D2
59
60 n(:) = n0(:)
61
62 t_end = 10.0D0
63 t_step = t_end / 1.0D10
64
65 Do k=1,10
66 DO i=1,1000000000
67 source = 0.0D0
68 loss = 0.0D0
69
70 source(1) = t_step * w(1)
71 loss(1) = n(1) * (1.0D0 - exp(-l(1) * t_step))
72
73 source(2) = t_step * w(2) + loss(1)
74 loss(2) = n(2) * (1.0D0 - exp(-l(2) * t_step))
75
76 source(3) = t_step * w(3) + loss(2)
77 loss(3) = n(3) * (1.0D0 - exp(-l(3) * t_step))
78
79 source(4) = t_step * w(4) + 0.359300D0 * loss(3)
80 loss(4) = n(4) * (1.0D0 - exp(-l(4) * t_step))
81
82 source(5) = t_step * w(5) + 0.640700D0 * loss(3)
83 loss(5) = n(5) * (1.0D0 - exp(-l(5) * t_step))
84
85 source(6) = t_step * w(6) + loss(4) + loss(5)
86
87 n(1:chn) = n(1:chn) + source(1:chn) - loss(1:chn)
88
89 END DO
90 END DO
91
92
93 print *, "Po216 ", n(1)
94 print *, "Pb212 ", n(2)
95 print *, "Bi212 ", n(3)
96 print *, "Tl208 ", n(4)
97 print *, "Po212 ", n(5)
98 print *, "Pb208 ", n(6)
99
100

```

```
101 END SUBROUTINE main  
102  
103  
104  
105 END PROGRAM main_test
```

### B.3.2 Chromium-49

Chromium-49 has a straightforward decay path, via positron emission (or electron capture) through to Vanadium-49 then to the stable isotope Titanium-49. Calculated isotope amounts are given for 5 different trial source rates and starting amounts; the results are listed in table ???. The RSS error between the numeric and analytic code in this decay chain over all five trials was  $6.52 \times 10^{-3}$  and the total percentage error between the two was  $2.17 \times 10^{-3}\%$ .

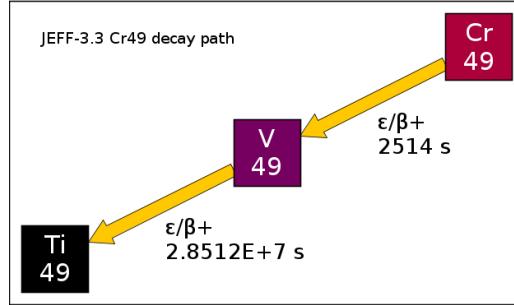


Figure B.1: Decay path for Chromium-49 [22]

		Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
$^{49}_{24}Cr$	$\omega$	$0.0 \times 10^0$	$1.0 \times 10^2$	$0.0 \times 10^0$	$2.5 \times 10^0$	$2.5 \times 10^0$
	$n_{start}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.0 \times 10^2$	$3.0 \times 10^4$	$3.0 \times 10^4$
	Numeric $n_{end}$	$0.0 \times 10^0$	$3.62694 \times 10^5$	$4.51169 \times 10^{-9}$	$9.06734 \times 10^3$	$9.06734 \times 10^3$
	Analytic $n_{end}$	$0.0 \times 10^0$	$3.62694 \times 10^5$	$4.51169 \times 10^{-9}$	$9.06734 \times 10^3$	$9.06734 \times 10^3$
$^{49}_{23}V$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.07 \times 10^0$
	$n_{start}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.0 \times 10^3$
	Numeric $n_{end}$	$0.0 \times 10^0$	$8.26897 \times 10^6$	$9.97990 \times 10^1$	$2.36664 \times 10^5$	$3.30013 \times 10^5$
	Analytic $n_{end}$	$0.0 \times 10^0$	$8.26897 \times 10^6$	$9.97990 \times 10^1$	$2.36664 \times 10^5$	$3.30013 \times 10^5$
$^{49}_{22}Ti$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$3.2 \times 10^0$
	$n_{start}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$2.0 \times 10^4$
	Numeric $n_{end}$	$0.0 \times 10^0$	$8.33841 \times 10^3$	$2.01023 \times 10^{-1}$	$2.68767 \times 10^2$	$2.96848 \times 10^5$
	Analytic $n_{end}$	$0.0 \times 10^0$	$8.33847 \times 10^3$	$2.01025 \times 10^{-1}$	$2.68769 \times 10^2$	$2.96848 \times 10^5$

Table B.2

### B.3.3 Nickel-66

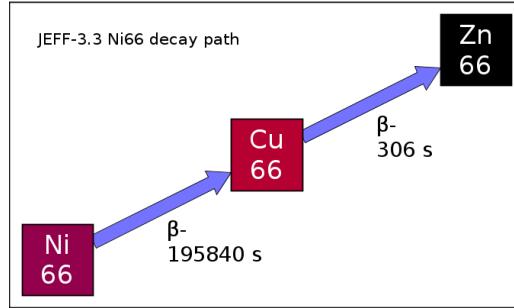


Figure B.2: Decay path for Nickel-66 [22]

Nickel-66 beta decays to Copper-66 and then to Zinc-66. The trial data are in table ???. The RSS error between the numeric and analytic code in this decay chain over all five trials was  $9.40 \times 10^{-2}$  and the total percentage error between the two was  $1.52 \times 10^{-4}\%$ .

		Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
$^{66}_{28}Ni$	$\omega$	$0.0 \times 10^0$	$1.0 \times 10^2$	$0.0 \times 10^0$	$2.5 \times 10^0$	$2.5 \times 10^0$
	$n_{\text{start}}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.0 \times 10^2$	$3.0 \times 10^4$	$3.0 \times 10^4$
Numeric	$n_{\text{end}}$	$0.0 \times 10^0$	$7.44391 \times 10^6$	$7.36534 \times 10^1$	$2.08194 \times 10^5$	$2.08194 \times 10^5$
Analytic	$n_{\text{end}}$	$0.0 \times 10^0$	$7.44391 \times 10^6$	$7.36534 \times 10^1$	$2.08194 \times 10^5$	$2.08194 \times 10^5$
$^{66}_{29}Cu$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.07 \times 10^0$
	$n_{\text{start}}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.0 \times 10^3$
Numeric	$n_{\text{end}}$	$0.0 \times 10^0$	$1.15802 \times 10^4$	$1.15264 \times 10^{-1}$	$3.24085 \times 10^2$	$7.96452 \times 10^2$
Analytic	$n_{\text{end}}$	$0.0 \times 10^0$	$1.15802 \times 10^4$	$1.15264 \times 10^{-1}$	$3.24085 \times 10^2$	$7.96452 \times 10^2$
$^{49}_{22}Ti$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$3.2 \times 10^0$
	$n_{\text{start}}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$2.0 \times 10^4$
Numeric	$n_{\text{end}}$	$0.0 \times 10^0$	$1.18451 \times 10^6$	$2.62314 \times 10^1$	$3.74822 \times 10^4$	$4.26938 \times 10^5$
Analytic	$n_{\text{end}}$	$0.0 \times 10^0$	$1.18451 \times 10^6$	$2.62314 \times 10^1$	$3.74822 \times 10^4$	$4.26938 \times 10^5$

Table B.4

### B.3.4 Caesium-125

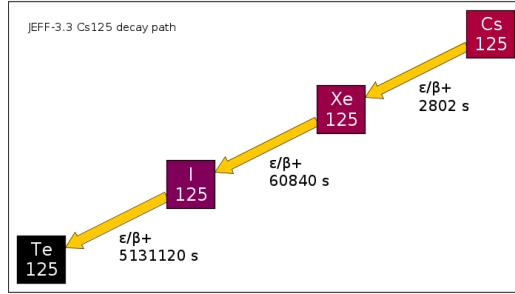


Figure B.3: Decay path for Caesium-125 [22]

The RSS error between the numeric and analytic code in this decay chain over all five trials was  $6.42 \times 10^{-1}$  and the total percentage error between the two was  $1.23 \times 10^{-3}\%$ .

		Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
$^{125}_{55}Cs$	$\omega$	$0.0 \times 10^0$	$1.0 \times 10^2$	$0.0 \times 10^0$	$5.0 \times 10^1$	$2.3 \times 10^1$
	$n_{\text{start}}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.0 \times 10^2$	$2.0 \times 10^5$	$3.0 \times 10^5$
Numeric	$n_{\text{end}}$	$0.0 \times 10^0$	$4.0424 \times 10^5$	$5.2204 \times 10^{-8}$	$2.0212 \times 10^5$	$9.2976 \times 10^4$
Analytic	$n_{\text{end}}$	$0.0 \times 10^0$	$4.0424 \times 10^5$	$5.2204 \times 10^{-8}$	$2.0212 \times 10^5$	$9.2976 \times 10^4$
$^{125}_{54}Xe$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$5.4 \times 10^1$
	$n_{\text{start}}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$2.0 \times 10^2$
Numeric	$n_{\text{end}}$	$0.0 \times 10^0$	$5.3391 \times 10^6$	$3.9172 \times 10^1$	$2.7479 \times 10^6$	$4.3142 \times 10^6$
Analytic	$n_{\text{end}}$	$0.0 \times 10^0$	$5.3391 \times 10^6$	$3.9172 \times 10^1$	$2.7479 \times 10^6$	$4.3142 \times 10^6$
$^{125}_{53}I$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$2.1 \times 10^1$
	$n_{\text{start}}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.7 \times 10^6$
Numeric	$n_{\text{end}}$	$0.0 \times 10^0$	$2.8851 \times 10^6$	$6.0438 \times 10^1$	$1.5634 \times 10^6$	$6.0191 \times 10^6$
Analytic	$n_{\text{end}}$	$0.0 \times 10^0$	$2.8851 \times 10^6$	$6.0438 \times 10^1$	$1.5634 \times 10^6$	$6.0191 \times 10^6$
$^{125}_{52}Te$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$4.0 \times 10^0$
	$n_{\text{start}}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$5.0 \times 10^4$
Numeric	$n_{\text{end}}$	$0.0 \times 10^0$	$1.1544 \times 10^4$	$3.8975 \times 10^{-1}$	$6.5513 \times 10^3$	$4.3678 \times 10^5$
Analytic	$n_{\text{end}}$	$0.0 \times 10^0$	$1.1544 \times 10^4$	$3.8975 \times 10^{-1}$	$6.5513 \times 10^3$	$4.3678 \times 10^5$

Table B.6

### B.3.5 Bismuth-213

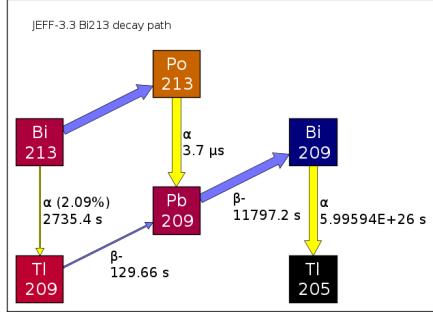


Figure B.4: Decay path for Bismuth-213 [22]

The previous decay examples have been straight chains with no branches, but Bismuth-213 is the first discussed in this section to have a branch. There's approximately an 98% chance that  $^{213}_{83}Bi$  will decay to  $^{213}_{83}Tl$  by alpha decay and just over 2% chance that it will decay to  $^{213}_{84}Po$  by beta decay. These both go on to decay to  $^{209}_{82}Pb$  that then undergoes beta decay to  $^{209}_{83}Bi$ . There is another step in the chain (to the stable isotope  $^{205}_{81}Tl$ ), but  $^{209}_{83}Bi$  is observationally stable with an extremely long half life and the decay code terminates the chain at this point.

The RSS error between the numeric and analytic code in this decay chain over all five trials was  $2.03 \times 10^{13}$ . This is large, but due mostly to the larger amounts used in this example. The total percentage error between the two was  $6.07 \times 10^3\%$  which is again large, but this is due to the numeric code and the short half life of Po-213 in comparison to the time step used. Without the contribution of Po-213 the total percentage error between the two sets of results is just  $2.37 \times 10^{-4}\%$ .

		Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
$^{213}_{83}Bi$	$\omega$	$0.0 \times 10^0$	$1.0 \times 10^2$	$0.0 \times 10^0$	$3.0 \times 10^5$	$3.0 \times 10^5$
	$n_{start}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.0 \times 10^0$	$1.0 \times 10^{10}$	$1.0 \times 10^{10}$
	Numeric	$n_{end}$	$0.0 \times 10^0$	$3.9463 \times 10^5$	$3.1024 \times 10^{-8}$	$1.1839 \times 10^9$
	Analytic	$n_{end}$	$0.0 \times 10^0$	$3.9463 \times 10^5$	$3.1024 \times 10^{-8}$	$1.1839 \times 10^9$
$^{213}_{84}Po$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$7.0 \times 10^3$
	$n_{start}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$2.0 \times 10^7$
	Numeric	$n_{end}$	$0.0 \times 10^0$	$8.4594 \times 10^{-3}$	$6.6504 \times 10^{-16}$	$2.5378 \times 10^1$
	Analytic	$n_{end}$	$0.0 \times 10^0$	$5.2264 \times 10^{-4}$	$4.1088 \times 10^{-17}$	$1.5679 \times 10^0$
$^{209}_{81}Tl$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$2.0 \times 10^4$
	$n_{start}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$2.0 \times 10^6$
	Numeric	$n_{end}$	$0.0 \times 10^0$	$3.9096 \times 10^2$	$3.2265 \times 10^{-11}$	$1.1729 \times 10^6$
	Analytic	$n_{end}$	$0.0 \times 10^0$	$3.9096 \times 10^2$	$3.2265 \times 10^{-11}$	$1.1729 \times 10^6$
$^{209}_{82}Pb$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.0 \times 10^0$
	$n_{start}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.0 \times 10^7$
	Numeric	$n_{end}$	$0.0 \times 10^0$	$1.6881 \times 10^6$	$8.1281 \times 10^{-1}$	$5.1457 \times 10^9$
	Analytic	$n_{end}$	$0.0 \times 10^0$	$1.6881 \times 10^6$	$8.1281 \times 10^{-1}$	$5.1457 \times 10^9$
$^{209}_{83}Bi$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$3.5 \times 10^8$
	$n_{start}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.0 \times 10^6$
	Numeric	$n_{end}$	$0.0 \times 10^0$	$6.5568 \times 10^6$	$9.9187 \times 10^1$	$2.9589 \times 10^{10}$
	Analytic	$n_{end}$	$0.0 \times 10^0$	$6.5568 \times 10^6$	$9.9187 \times 10^1$	$2.9589 \times 10^{10}$
$^{205}_{81}Tl$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.0 \times 10^{-10}$
	$n_{start}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.0 \times 10^{-10}$
	Numeric	$n_{end}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$
	Analytic	$n_{end}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$

Table B.8

### B.3.6 Polonium-216

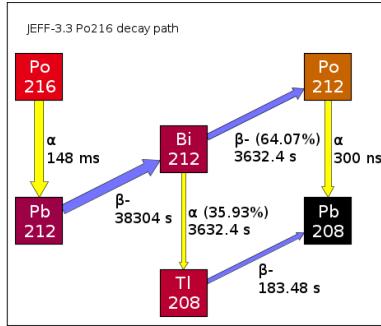


Figure B.5: Decay path for Bismuth-213 [22]

The RSS error between the numeric and analytic code in this decay chain over all five trials was  $3.85 \times 10^1$  and the total percentage error between the two was  $7.94 \times 10^4\%$ . This large error was due to a short half life isotope (Po-212) and the comparatively long time step in the numeric solver. Without this result, the agreement between numeric and analytic is  $6.10 \times 10^{-2}\%$ .

		Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
$^{216}_{84}Po$	$\omega$	$0.0 \times 10^0$	$1.0 \times 10^2$	$0.0 \times 10^0$	$3.5 \times 10^2$	$2.0 \times 10^{-1}$
	$n_{start}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.0 \times 10^2$	$5.0 \times 10^4$	$1.0 \times 10^2$
	$n_{end}$	$0.0 \times 10^0$	$2.13562 \times 10^1$	$0.0 \times 10^0$	$7.47467 \times 10^1$	$4.27124 \times 10^{-2}$
	Analytic	$0.0 \times 10^0$	$2.13519 \times 10^1$	$0.0 \times 10^0$	$7.47316 \times 10^1$	$4.27038 \times 10^{-2}$
$^{212}_{82}Pb$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$2.0 \times 10^{-1}$
	$n_{start}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$5.0 \times 10^0$
	Numeric	$0.0 \times 10^0$	$4.36891 \times 10^6$	$2.09405 \times 10^1$	$1.53016 \times 10^7$	$1.31287 \times 10^4$
	Analytic	$0.0 \times 10^0$	$4.36891 \times 10^6$	$2.09405 \times 10^1$	$1.53017 \times 10^7$	$1.31287 \times 10^4$
$^{212}_{83}Bi$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$7.0 \times 10^{-2}$
	$n_{start}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.50 \times 10^1$
	Numeric	$0.0 \times 10^0$	$4.02810 \times 10^5$	$2.19385 \times 10^0$	$1.41093 \times 10^6$	$1.57757 \times 10^3$
	Analytic	$0.0 \times 10^0$	$4.02810 \times 10^5$	$2.19385 \times 10^0$	$1.41093 \times 10^6$	$1.57757 \times 10^3$
$^{208}_{81}Tl$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$5.0 \times 10^{-3}$
	$n_{start}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.0 \times 10^1$
	Numeric	$0.0 \times 10^0$	$7.30001 \times 10^3$	$4.00077 \times 10^{-2}$	$2.55700 \times 10^4$	$1.83923 \times 10^{-5}$
	Analytic	$0.0 \times 10^0$	$\times 10$	$\times 10$	$\times 10$	$\times 10$
$^{212}_{84}Po$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$2.0 \times 10^{-2}$
	$n_{start}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.7 \times 10^1$
	Numeric	$0.0 \times 10^0$	$4.25501 \times 10^{-3}$	$2.31743 \times 10^{-8}$	$1.49041 \times 10^{-2}$	$1.83923 \times 10^{-5}$
	Analytic	$0.0 \times 10^0$	$2.13149 \times 10^{-5}$	$1.16088 \times 10^{-10}$	$7.46601 \times 10^{-5}$	$9.21338 \times 10^{-8}$
$^{208}_{82}Pb$	$\omega$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$1.0 \times 10^{-2}$
	$n_{start}$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$0.0 \times 10^0$	$3.0 \times 10^2$
	Numeric	$0.0 \times 10^0$	$3.86096 \times 10^6$	$7.68257 \times 10^1$	$1.35518 \times 10^7$	$2.07028 \times 10^4$
	Analytic	$0.0 \times 10^0$	$3.86096 \times 10^6$	$7.68257 \times 10^1$	$1.35518 \times 10^7$	$2.07028 \times 10^4$

Table B.10

### B.3.7 Radon-218

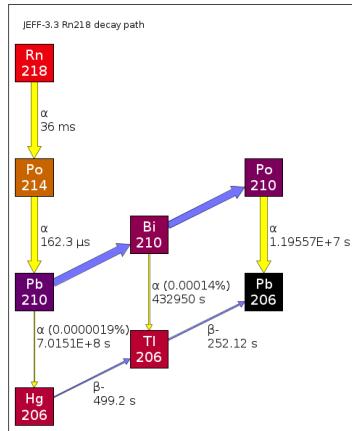


Figure B.6: Decay path for Bismuth-213 [22]

The RSS error between the numeric and analytic code in this decay chain over all five trials was  $1.73 \times 10^3$  and the total percentage error between the two was  $5.93 \times 10^1\%$ . However, if the short half life isotope Po-214 is excluded, the error is just  $5.15 \times 10^{-1}\%$ .

		Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
$^{218}_{86}Rn$	$\omega$	$0.0 \times 10^0$	$1.0000 \times 10^0$	$0.0000 \times 10^0$	$3.5 \times 10^2$	$3.5000 \times 10^2$
	$n_{\text{start}}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$1.0000 \times 10^0$	$5.0000 \times 10^4$	$5.0000 \times 10^4$
	Numeric	$0.0000 \times 10^0$	$5.1980 \times 10^0$	$0.0000 \times 10^0$	$1.8193 \times 10^1$	$1.8193 \times 10^1$
	Analytic	$0.0000 \times 10^0$	$5.1937 \times 10^0$	$0.0000 \times 10^0$	$1.8178 \times 10^1$	$1.8178 \times 10^1$
$^{214}_{84}Po$	$\omega$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$7.0000 \times 10^1$
	$n_{\text{start}}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$1.0000 \times 10^3$
	Numeric	$0.0000 \times 10^0$	$2.8000 \times 10^{-2}$	$0.0000 \times 10^0$	$9.8000 \times 10^{-2}$	$1.1760 \times 10^{-1}$
	Analytic	$0.0000 \times 10^0$	$2.3415 \times 10^{-2}$	$0.0000 \times 10^0$	$8.1952 \times 10^{-2}$	$9.8343 \times 10^{-1}$
$^{210}_{82}Pb$	$\omega$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$1.0000 \times 10^3$
	$n_{\text{start}}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$1.2000 \times 10^2$
	Numeric	$0.0000 \times 10^0$	$8.6396 \times 10^6$	$9.9991 \times 10^1$	$3.0289 \times 10^7$	$1.2273 \times 10^8$
	Analytic	$0.00000 \times 10^0$	$8.6396 \times 10^6$	$9.9991 \times 10^1$	$3.0289 \times 10^7$	$1.2273 \times 10^8$
$^{206}_{80}Hg$	$\omega$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$1.7000 \times 10^1$
	$n_{\text{start}}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$7.0000 \times 10^2$
	Numeric	$0.0000 \times 10^0$	$1.1585 \times 10^{-7}$	$1.3520 \times 10^{-12}$	$4.0614 \times 10^{-7}$	$1.2243 \times 10^4$
	Analytic	$0.00000 \times 10^0$	$1.1584 \times 10^{-7}$	$1.3519 \times 10^{-12}$	$4.0611 \times 10^{-7}$	$1.2243 \times 10^4$
$^{210}_{83}Bi$	$\omega$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$2.3000 \times 10^1$
	$n_{\text{start}}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$4.1000 \times 10^1$
	Numeric	$0.0000 \times 10^0$	$3.5238 \times 10^2$	$7.9731 \times 10^{-3}$	$1.2373 \times 10^3$	$1.8609 \times 10^{-6}$
	Analytic	$0.00000 \times 10^0$	$3.5236 \times 10^2$	$7.9725 \times 10^{-3}$	$1.2372 \times 10^3$	$1.8609 \times 10^6$
$^{206}_{81}Tl$	$\omega$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$8.1000 \times 10^1$
	$n_{\text{start}}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$5.0000 \times 10^2$
	Numeric	$0.0000 \times 10^0$	$3.4319 \times 10^{-7}$	$7.1575 \times 10^{-12}$	$1.2047 \times 10^{-6}$	$3.5646 \times 10^4$
	Analytic	$0.00000 \times 10^0$	$3.4316 \times 10^{-7}$	$7.1570 \times 10^{-12}$	$1.2046 \times 10^{-6}$	$3.5646 \times 10^4$
$^{210}_{84}Po$	$\omega$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$5.2000 \times 10^2$
	$n_{\text{start}}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$1.5000 \times 10^3$
	Numeric	$0.0000 \times 10^0$	$1.6413 \times 10^1$	$5.6320 \times 10^{-4}$	$5.7726 \times 10^1$	$4.4948 \times 10^7$
	Analytic	$0.00000 \times 10^0$	$1.6411 \times 10^1$	$5.6316 \times 10^{-4}$	$5.7722 \times 10^1$	$4.4948 \times 10^7$
$^{206}_{82}Pb$	$\omega$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$1.0000 \times 10^3$
	$n_{\text{start}}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$1.0000 \times 10^4$
	Numeric	$0.0000 \times 10^0$	$2.0729 \times 10^{-2}$	$9.5248 \times 10^{-7}$	$7.3026 \times 10^{-2}$	$9.4943 \times 10^7$
	Analytic	$0.00000 \times 10^0$	$2.0747 \times 10^{-2}$	$9.5242 \times 10^{-7}$	$7.3089 \times 10^{-2}$	$9.4943 \times 10^7$

Table B.12

# Appendix C

## Activity Code

### C.1 Activity V1

#### C.1.1 Paper published in Computer Physics Communications

The Activity code was published in the Computer Physics Communications journal and this is included in the following pages. It was submitted in 2015 and published in 2017.

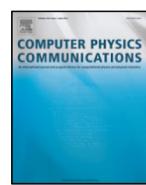
Computer Physics Communications 216 (2017) 138–144



Contents lists available at ScienceDirect

## Computer Physics Communications

journal homepage: [www.elsevier.com/locate/cpc](http://www.elsevier.com/locate/cpc)



# Activity computer program for calculating ion irradiation activation<sup>☆</sup>

Ben Palmer <sup>\*</sup>, 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](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 exyz 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:

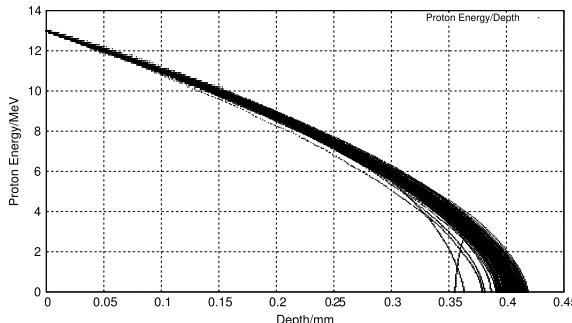
- [1] SRIM – The stopping and range of ions in matter (2010). Ziegler, James F., Ziegler, M.D. and Biersack, J.P. 2010, Nuclear Instruments and Methods in Physics Research Section B, Vol. 268, pp. 1818–1823.

© 2017 Elsevier B.V. All rights reserved.

<sup>☆</sup> 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](mailto: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  $\mu\text{A}$  ( $3.1 \times 10^{14}$  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)
- $\sigma$  Microscopic reaction cross section (barns)
- e Elementary charge (1.602177E-19C)
- $\delta t$  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,  $\lambda$ , 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:

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

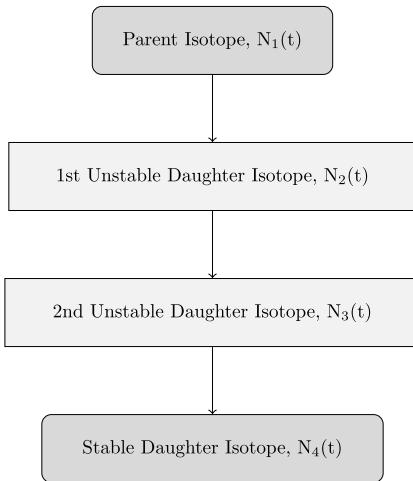
$$A(t) = \lambda N(0) \exp(-t\lambda). \quad (5)$$

### 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  $\lambda_i$
- $\vec{b}$  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 ith isotope

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

The differential equations for the parent isotope (first isotope), unstable daughter isotopes (ith isotopes) and stable, final, daughter isotope (zth 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).

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

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

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

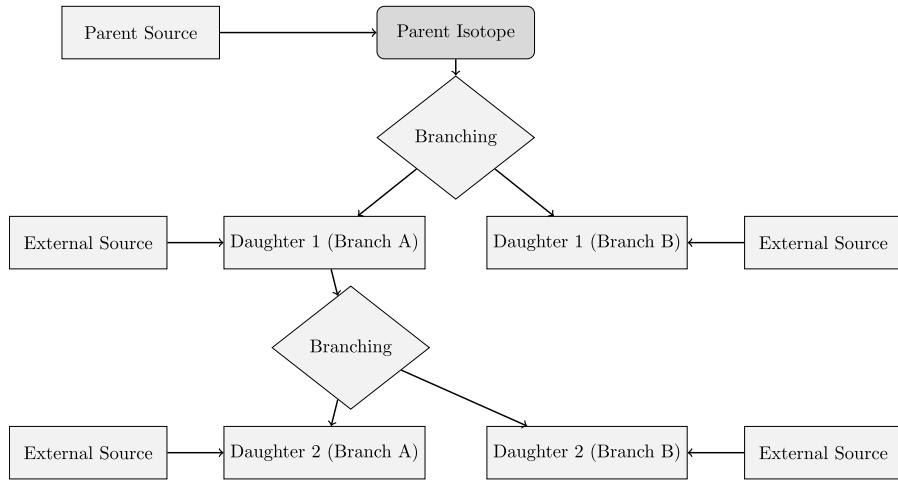
The equation for the ith isotope may be calculated by recursively calculating the equations by numeric inversion, starting from the first (parent isotope) and inserting the result into each subsequent recursion until the ith isotope is reached (changing the equations appropriately for the parent, unstable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

### 1.9. Analytic solution by partial fraction expansion

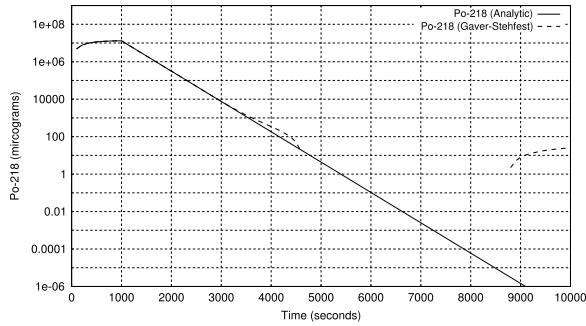
The equation for the ith isotope in the s domain can be written in full by substituting the preceding equation until the parent isotope is reached, and this full equation may be rearranged with the production amount of each isotope and starting amount of each isotope in individual terms. Each of these 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.

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

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

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

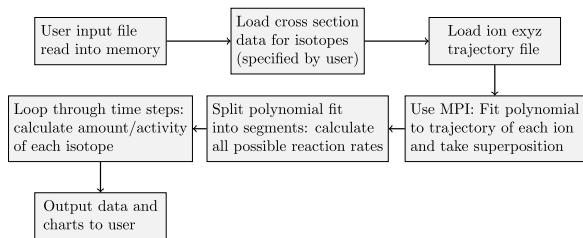
$$g(t; k, m, \vec{\lambda}) = \frac{1}{\prod_{i=k, m} \lambda_i} + (-1)^{m-k+1} \times \sum_{i=k, m} \left[ \frac{1}{\lambda_i} \exp(-\lambda_i t) \prod_{j=k, m; j \neq i} \left( \frac{1}{\lambda_i - \lambda_j} \right) \right]. \quad (22)$$

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.

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

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

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



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

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

### 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 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 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

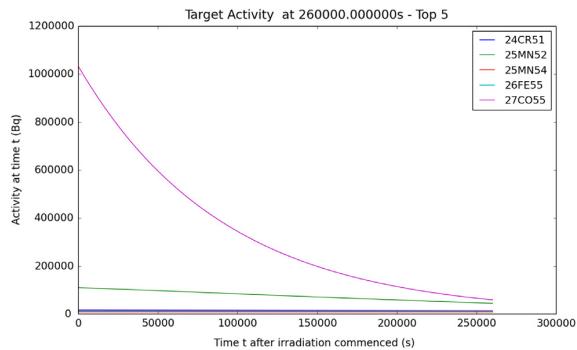
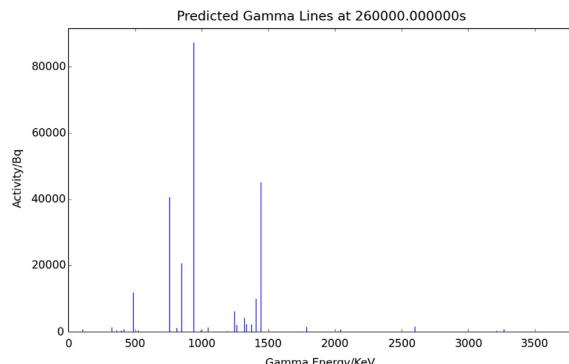
A target of high purity Iron was irradiated with 36 MeV protons by the University of Birmingham Scanditronix MC-40 Cyclotron. The target was 0.5 mm thick and was irradiated at a current of 0.5  $\mu$ A for 300 s, irradiating approximately 0.25 g of Iron. A high purity Germanium detector was used to measure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

**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.

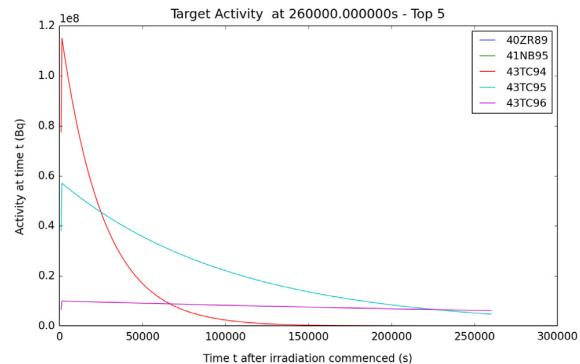
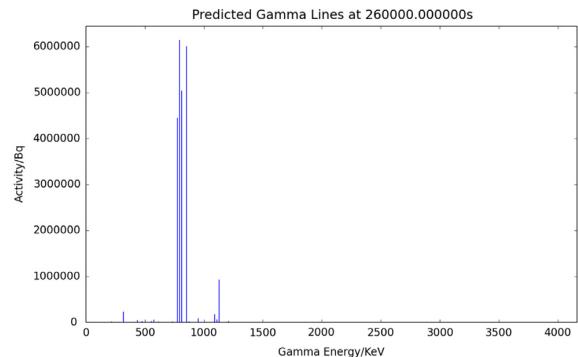
A target of high purity Molybdenum was irradiated with 13 MeV protons by the University of Birmingham Scanditronix MC-40 Cyclotron. The target was 0.5 mm thick and was irradiated at a current of 5  $\mu$ A for 1500 s, irradiating approximately 0.3 g of Molybdenum. A high purity Germanium detector was 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/ndspub/download-endf/ENDF-B-VII.0/decay-index.htm](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.

### C.1.2 Accompanying Manual

The accompanying manual for the Activity code is included in the following pages.

# UNIVERSITY OF BIRMINGHAM



Department of Metallurgy & Materials

Activity Manual

# Contents

<b>1 Background</b>	<b>4</b>
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
<b>2 Installation and Using the Activity Code</b>	<b>17</b>
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
<b>Appendices</b>	<b>22</b>

<b>A Example Input File</b>	<b>23</b>
A.1 Iron 36MeV Proton Beam . . . . .	23
<b>B Fortran 90 Code</b>	<b>25</b>
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: Analytice 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 ( $3.1 \times 10^{14}$  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.

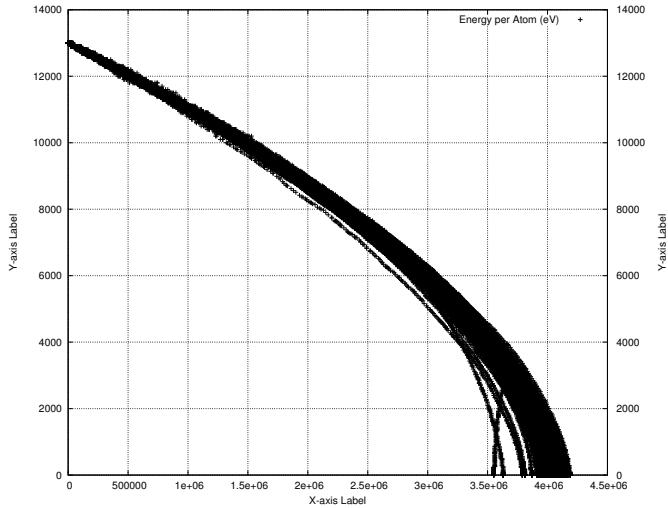


Figure 1.1: Proton energy loss in Fe simulated with SRIM [2]

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)
- $\sigma$  Microscopic reaction cross section (barns)
- e Elementary charge (1.602177E-19C)
- $\delta T$  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  $\lambda$  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

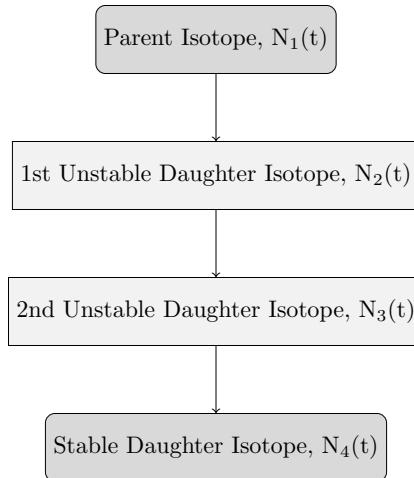


Figure 1.2: An example decay chain from an unstable parent isotope, through unstable daughter isotopes ending with a stable daughter isotope.

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  $t$ .

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

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.

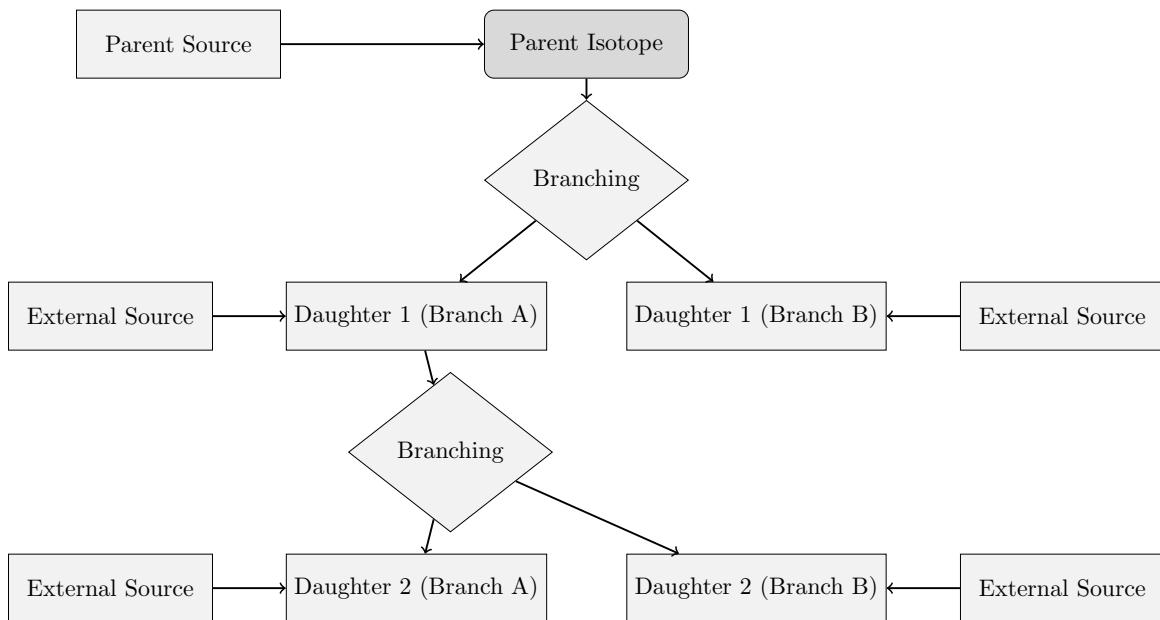


Figure 1.3: An example of several decay chains including branching factors and possible external source terms for each isotope on each chain.

### 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  $\lambda_i$
- $\vec{b}$  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 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):

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

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 mth isotope in the decay chain, providing the mth 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 mth isotope in the decay chain, where the mth isotope is stable.

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

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

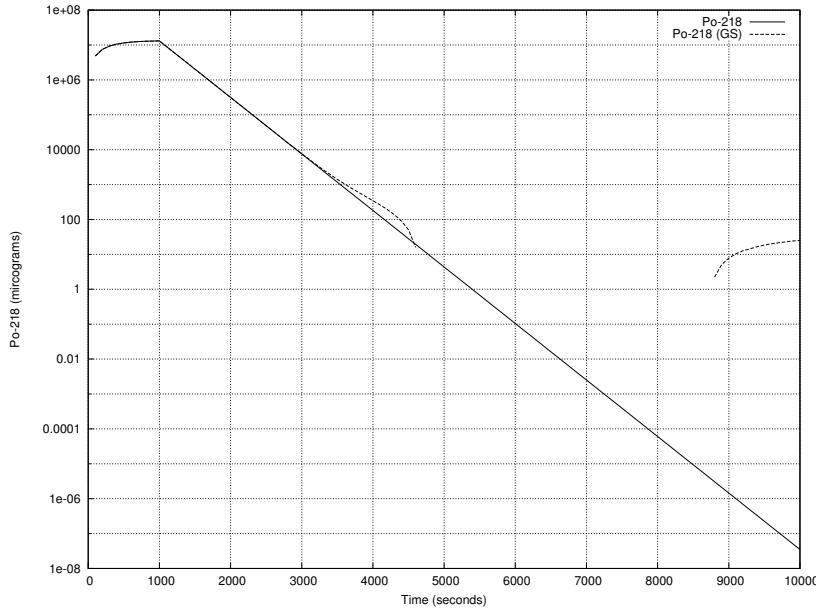


Figure 1.4: Decay of Po-218: Analytic and Gaver-Stehfest Calculations [5]

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

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

### 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.

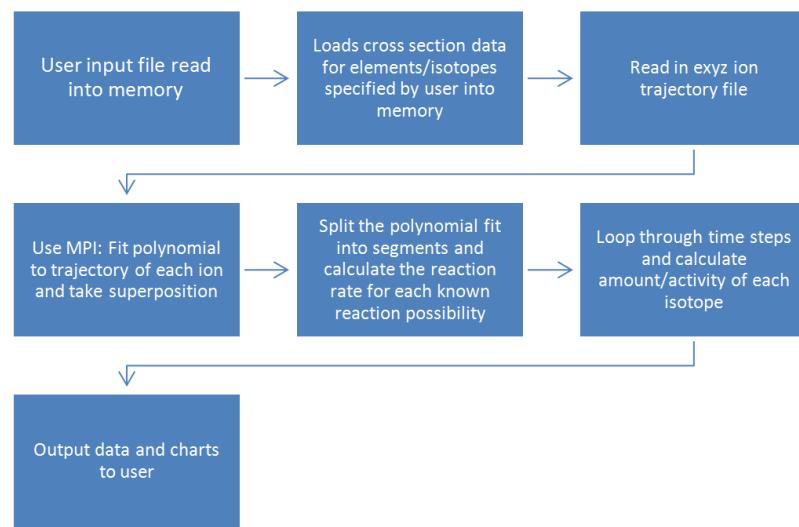


Figure 1.5: Flow chart of major processes in the Activity code

## 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

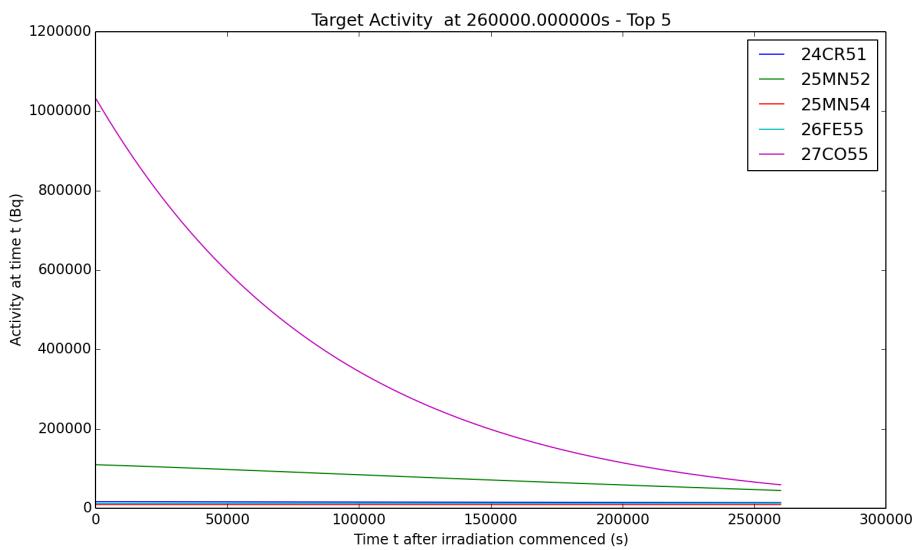


Figure 1.6: Sample Activity code output chart for the top five most active isotopes for Iron irradiated by 36MeV protons.

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.

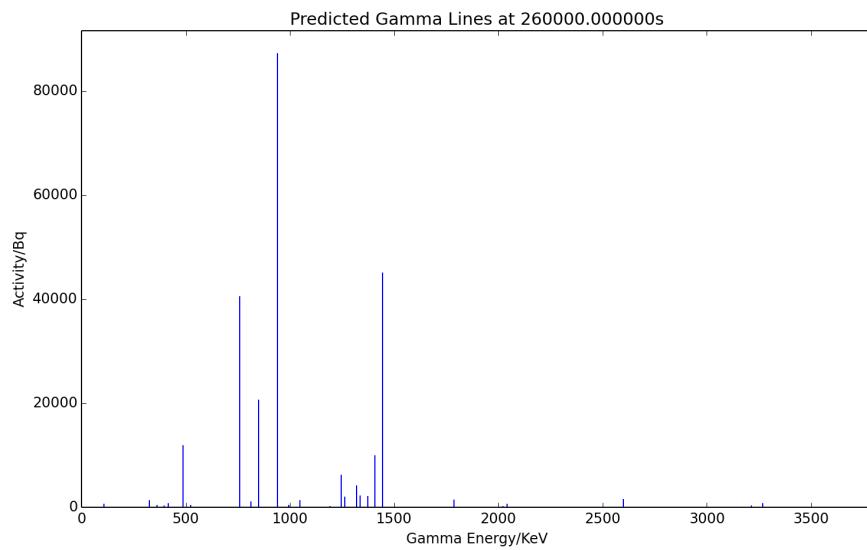


Figure 1.7: Sample Activity code output chart for the expected gamma lines to be measured for Iron irradiated by 36MeV protons.

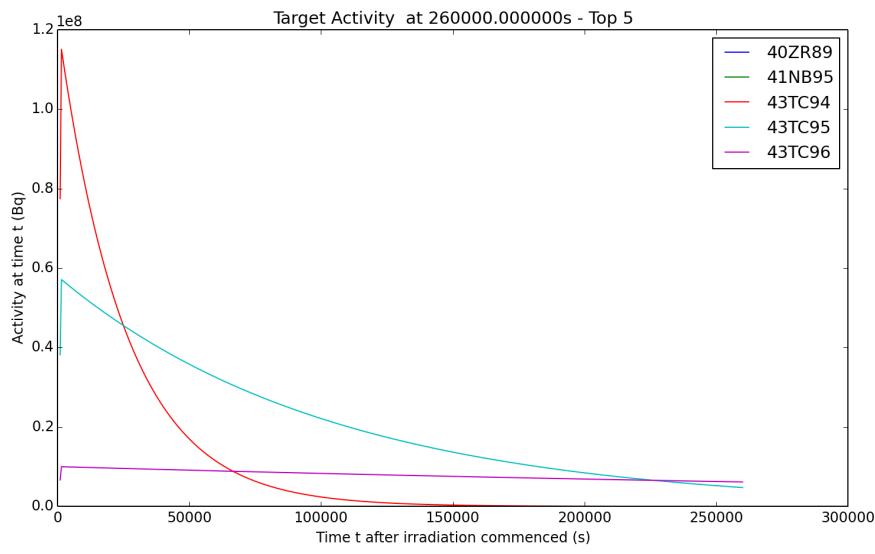


Figure 1.8: Sample Activity code output chart for the top five most active isotopes for Molybdenum irradiated by 13MeV protons.

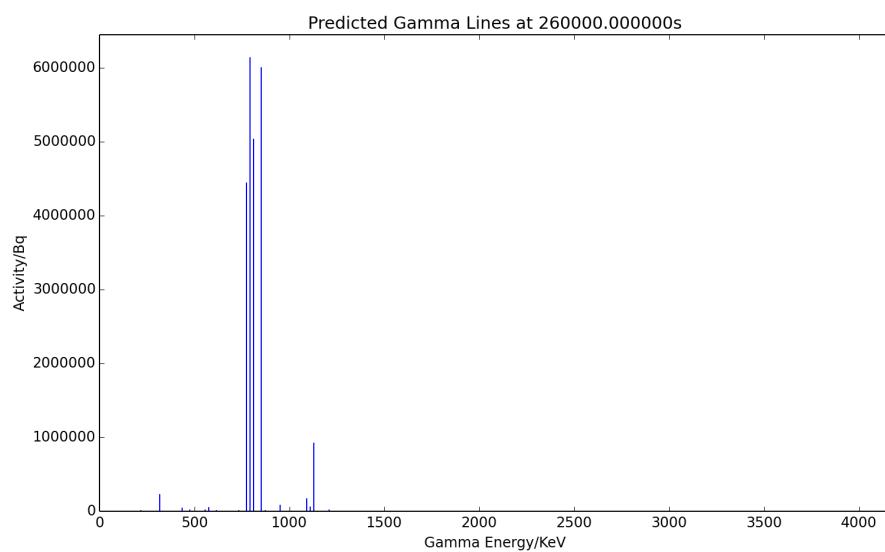


Figure 1.9: Sample Activity code output chart for the expected gamma lines to be measured for Molybdenum irradiated by 13MeV protons.

## 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 ommited 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 calulcation. #verboseterminal followed by yeswill 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 ommited 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 #beadmduration
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/ndspub/download-endf/ENDF-B-VII.0/decay-index.htm](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.

## C.2 Activity V2

### C.2.1 Accompanying Manual

# UNIVERSITY OF BIRMINGHAM



Department of Metallurgy & Materials

**Activity V2 Manual**

# Contents

<b>1 Overview</b>	<b>2</b>
1.1 Reason for the Program . . . . .	2
1.2 Program Requirements . . . . .	2
<b>2 Installation</b>	<b>3</b>
<b>3 How To Use</b>	<b>4</b>
3.1 SRIM . . . . .	4
3.2 Activity . . . . .	5
3.3 Input Keywords . . . . .	11
<b>4 Examples</b>	<b>12</b>
4.1 Iron - 28MeV Protons . . . . .	12
4.2 Iron Fe56 (only) - 28MeV Protons . . . . .	12
4.3 Steel - 5MeV Protons . . . . .	12
<b>5 Decay Equation</b>	<b>14</b>
5.1 Bateman Equation . . . . .	14
5.1.1 Laplace Transform . . . . .	14
5.1.2 Constructing the Differential Equations . . . . .	15
5.1.3 Numerical Inversion of the Laplace Transform . . . . .	15
5.1.4 Analytic Solution by Partial Fraction Expansion . . . . .	16
5.1.5 Preference: Analytic over Numeric . . . . .	18
5.2 Python Isotopes Class . . . . .	19
<b>6 Future Plans</b>	<b>22</b>
6.1 Selection of Cross Section Databases . . . . .	22

# Chapter 1

## Overview

### 1.1 Reason for the Program

The Activity program calculates how radioactive a target becomes after being irradiated by high energy ions. It uses the TENDL-2019[1] database that contains cross-section data for protons and deuterons, and the JEFF-3.3 Radioactive Decay Data File[2].

### 1.2 Program Requirements

The user must provide a exyz file from the SRIM[3] ion transport program and a breakdown of the composition of the target, as well as other irradiation parameters (beam projectile, beam duration, beam area, target thickness, target composition, simulation end time etc).

# Chapter 2

## Installation

The program needs Python 3 installed in order to run. At the time of writing, it has only been developed to run on a Linux operating system, but it shouldn't require much adjusting to run on a Windows computer too.

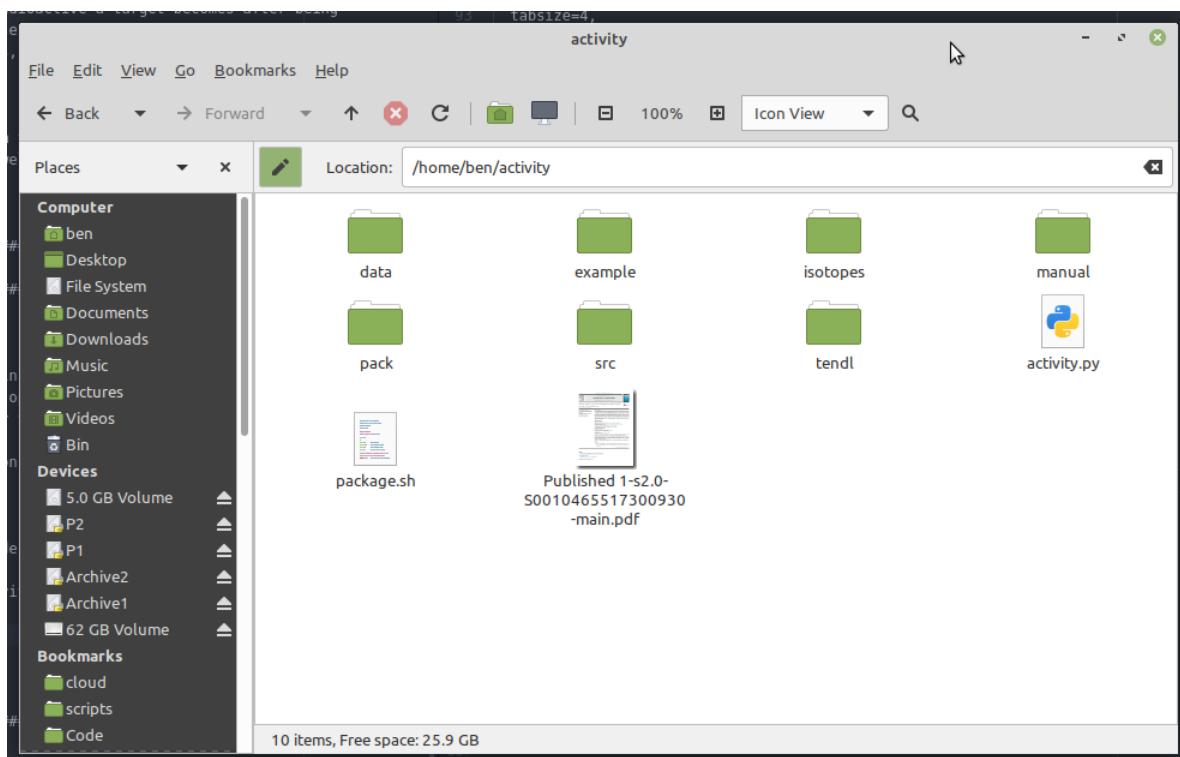
Download and install the latest version of Python 3:

<https://www.python.org/downloads>

The latest version of the activity code with data files must also be downloaded:

[https://github.com/BenPalmer1983/activity\\_v2](https://github.com/BenPalmer1983/activity_v2)

For example, I have installed to /home/ben/activity

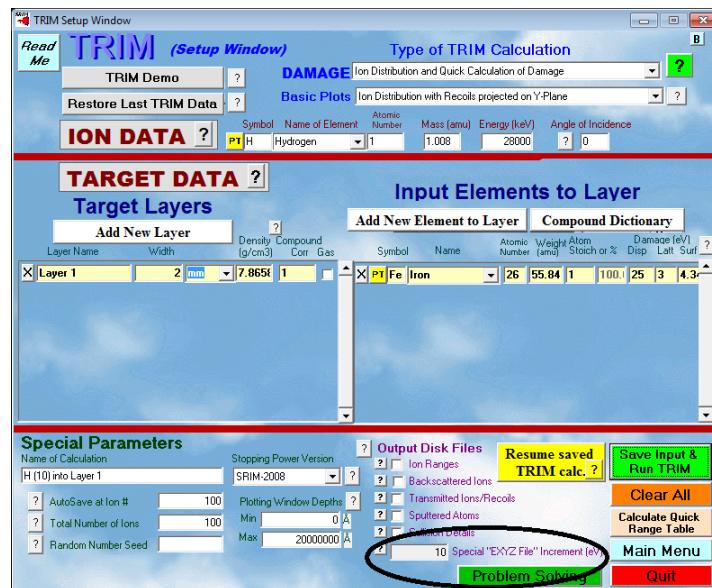


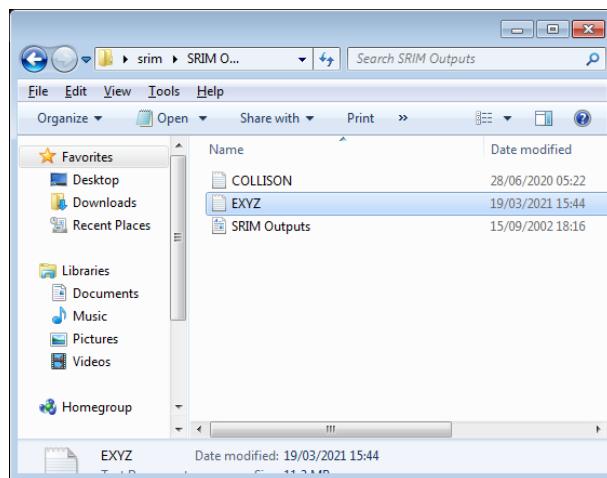
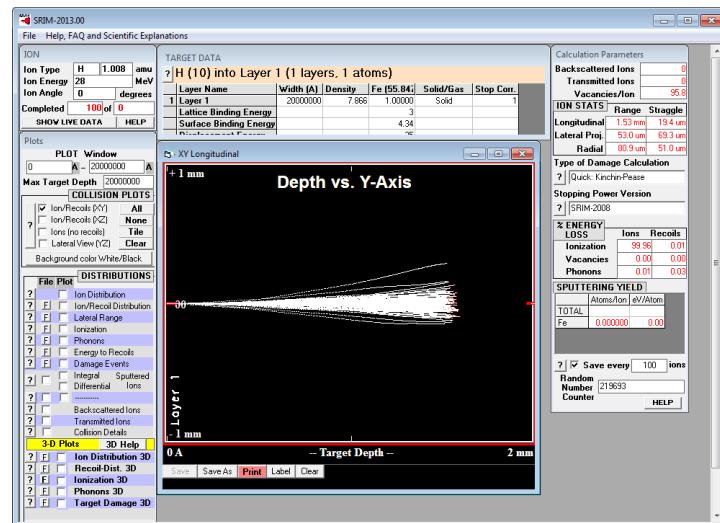
# Chapter 3

## How To Use

### 3.1 SRIM

If you haven't done so already, run the required simulation in SRIM. When setting up the calculation, be sure to set an increment for the EXYZ file. Sane values that have been used in examples range from 10eV to 10,000KeV, but this will depend on the resolution of the data in the TENDL database, the thickness of your target and the energy of the projectiles.

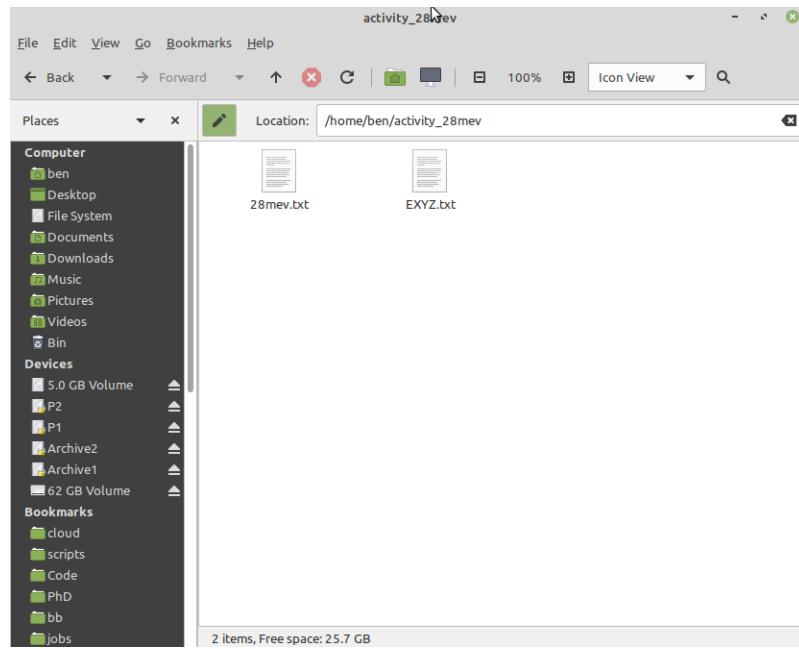




Take a copy of the EXYZ.txt file as it will be required by the activity program.

## 3.2 Activity

Create a directory to run the calculation. Copy in the EXYZ.txt file and create an input file.



The input file is just a text file, and will specify the calculation details. The data file paths must be specified, and then the simulations can be detailed, each with it's own unique name. The example below is for two simulations - the first with a 300s beam, and the second with a 3000s beam.

---

```

1 # Data files
2 data_isotopes="/home/ben/activity/data/isotopes" xs="/home/ben/activity/data/xs"
3
4 # Sim 1
5 sim1 exyz="EXYZ.txt" target_composition=Fe,100.0 target_depth=0.5,mm target_density=7808,kgm3
   beam_projectile='proton' beam_energy=28,MeV beam_area=64,mm2 beam_duration=300,s beam_current=0.5,
   uA end_time=260000,s
6
7 # Sim 2
8 sim2 exyz="EXYZ.txt" target_composition=Fe,100.0 target_depth=0.5,mm target_density=7808,kgm3
   beam_projectile='proton' beam_energy=28,MeV beam_area=64,mm2 beam_duration=3000,s beam_current
   =5.0,uA end_time=260000,s

```

---

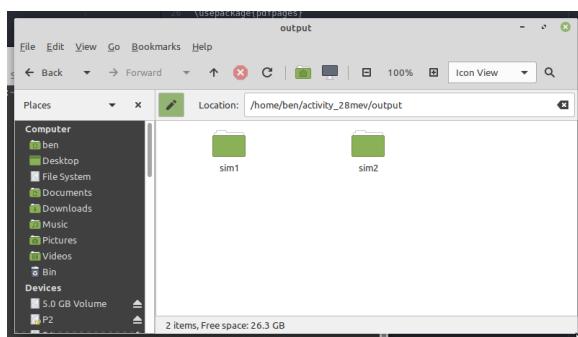
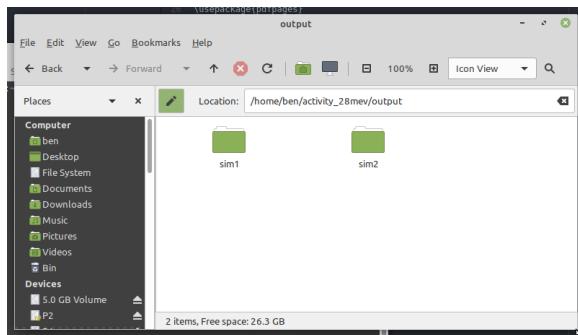
The program is run from the terminal/command line. Depending on your computer system, python3 might run through the command python3 or python, and on my computer the command is python3.

---

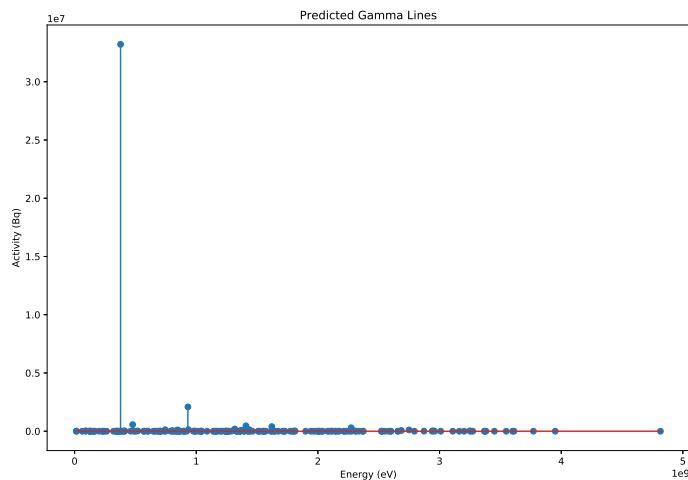
```
1 python3 /home/ben/activity/activity.py /home/ben/activity_28mev/28mev.txt
```

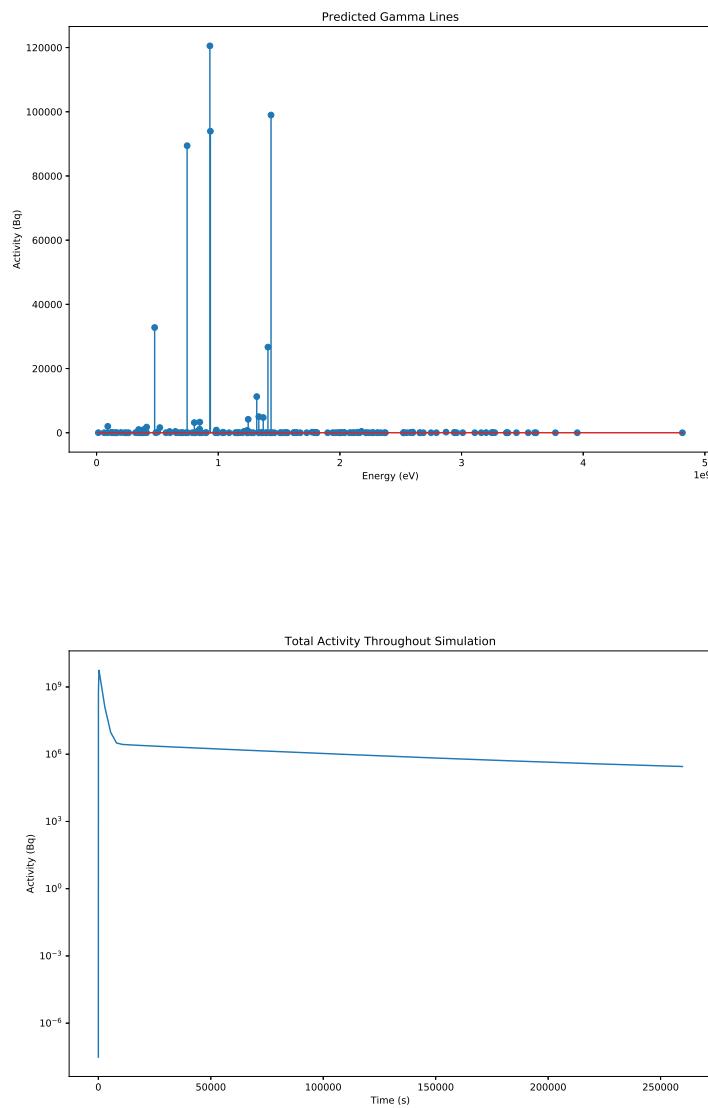
---

The program outputs into a directory for each simulation.



Within each directory are various plots and data files, including the activity over time and predicted gamma lines.





The program analyses the exyz file and output a chart showing the distribution of energy lost by projectiles as they pass through the target.

As a target is irradiated, the residual radioactive isotopes begin to decay. To begin with, the production outweighs the decay, but at a certain point there's enough of the radioactive isotope within the target that the decay balances with the production. At this point the radioactive isotope in the target is saturated, and will not increase above this amount unless the beam parameters are adjusted.

The saturation times depend on the isotope and the various reaction cross sections with the projectile. The saturation plots for each radioactive isotope are created and saved by the program.

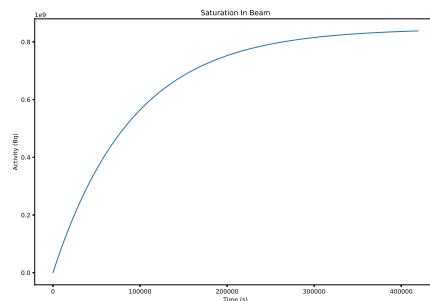
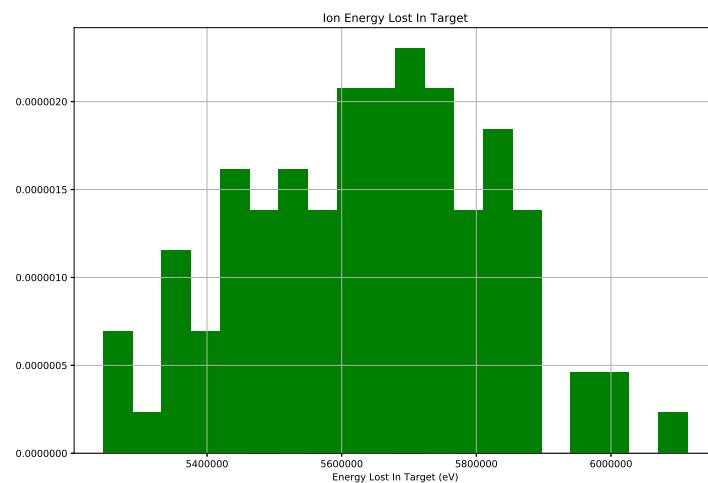


Figure 3.1: Saturation of Cobalt-55

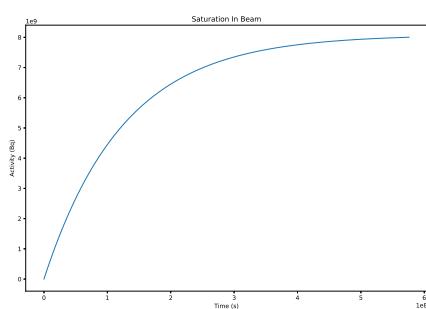


Figure 3.2: Saturation of Iron-55

The program calculates the dose a human may receive from the target at the time it is removed from the beam, and at the time the simulation stops.

Listing 3.1: Predicted Dose

---

```

1 Gamma Dose - Beam End
2 =====
3
4 Activity/Bq           54.4306601689
5 Power eV/s            28342871818.2
6 Power J/s             4.54103823697e-09
7 Dose Gy/s             4.51575003676e-12
8 Dose Gy/hr             1.62567001323e-08
9 Percentage of annual dose/hr 0.0142506237521

10
11 Gamma Dose - Sim End
12 =====
13
14 Activity/Bq           5.80221584491
15 Energy eV/s            2227911995.02
16 Energy J/s             3.56951604018e-10
17 Dose Gy                8.52319971389e-13
18 Dose Gy/hr              1.27786970412e-09
19 Percentage of annual dose/hr 0.00112018061534

20
21
22 Absorbed Dose Calculations
23 =====
24 Absorbed dose assumptions:
25 1. radiation from point, emitted isotropically
26 2. 80Kg human
27 3. 1m from point source
28 4. 1m squared surface area
29 5. all energy absorbed

30
31
32 Dose Limits
33 =====
34 employees 18+          20 millisieverts/year
35 trainees 18+            6 millisieverts/year
36 public and under 18s    1 millisievert/year
37 public and under 18s    1.140771128E-04 millisieverts/hour

38
39 Dose averaged over area of skin not exceeding 1cm2
40 Source: http://www.hse.gov.uk/radiation/ionising/doses/
```

---

### 3.3 Input Keywords

<b>Command</b>	<i>Sub Command</i>	Description
<b>data</b>		data fields are input here
<b>data</b>	isotopes	path to isotopes directory
<b>data</b>	xs	path to cross section directory
<b>sim1</b>		each simulation is numbered sequentially from 1 i.e. sim1 sim2 sim3...
<b>sim1</b>	exyz	path to exyz SRIM file
<b>sim1</b>	target_composition	isotopes that make up the beam target
<b>sim1</b>	target_depth	the depth of material through which the beam passes
<b>sim1</b>	target_density	how dense (specify kgm3 or GCM3)
<b>sim1</b>	beam_projectile	currently, only "proton", but this could be extended in the future
<b>sim1</b>	beam_energy	energy of the protons/projectiles
<b>sim1</b>	beam_area	how much area the beam covers
<b>sim1</b>	beam_duration	how long the beam will be on for (specify s,minute,hour) t=0 to beam_duration is the first part of the calculation
<b>sim1</b>	end_time	the time the calculation and final activity is measured after t=0 (specify s,minute,hour) end_time to beam_duration is the second part of the calculation

# Chapter 4

## Examples

### 4.1 Iron - 28MeV Protons

In this example, natural iron makes up the target .

---

```

1 # Data files
2 data isotopes="/home/ben/activity/data/isotopes" xs="/home/ben/activity/data/xs"
3
4 # Sim
5 sim1 exyz="XYZ.txt" target_composition=Fe,100.0 target_depth=0.5,mm target_density=7808,kgm3
  beam_projectile='proton' beam_energy=28,MeV beam_area=64,mm2 beam_duration=300,s beam_current=0.5,
  uA end_time=260000,s

```

---

The simulation section defines what the target material is made of as well as the beam parameters.

### 4.2 Iron Fe56 (only) - 28MeV Protons

In this example, the target is made of Fe56 only, and contains none of the other naturally occurring stable isotopes.

---

```

1 # Data files
2 data isotopes="/home/ben/activity/data/isotopes" xs="/home/ben/activity/data/xs"
3
4 # Sim
5 sim1 exyz="XYZ.txt" target_composition=Fe56,100.0 target_depth=0.5,mm target_density=7808,kgm3
  beam_projectile='proton' beam_energy=28,MeV beam_area=64,mm2 beam_duration=300,s beam_current=0.5,
  uA end_time=260000,s

```

---

### 4.3 Steel - 5MeV Protons

This example was provided by Alex Dickinson-Lomas, with a steel containing a wider range of elements.

---

```

1 # Data files
2 data isotopes="/home/ben/activity/data/isotopes" xs="/home/ben/activity/data/xs"
3

```

```
4 # Sim
5 sim1 exyz="EXYZ.txt" target_composition=Fe,96.375,C,0.772,Cu,0.024,Mn,1.36,Ni,0.698,Si,0.381,Cr,0.092,
V,0.008,P,0.009,Si,0.003,Mo,0.278 target_depth=0.1,mm target_density=7808,kgm3 beam_projectile='
proton' beam_energy=5,MeV beam_area=64,mm2 beam_duration=300,s beam_current=0.5,uA end_time
=260000,s
```

# Chapter 5

## Decay Equation

### 5.1 Bateman Equation

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.

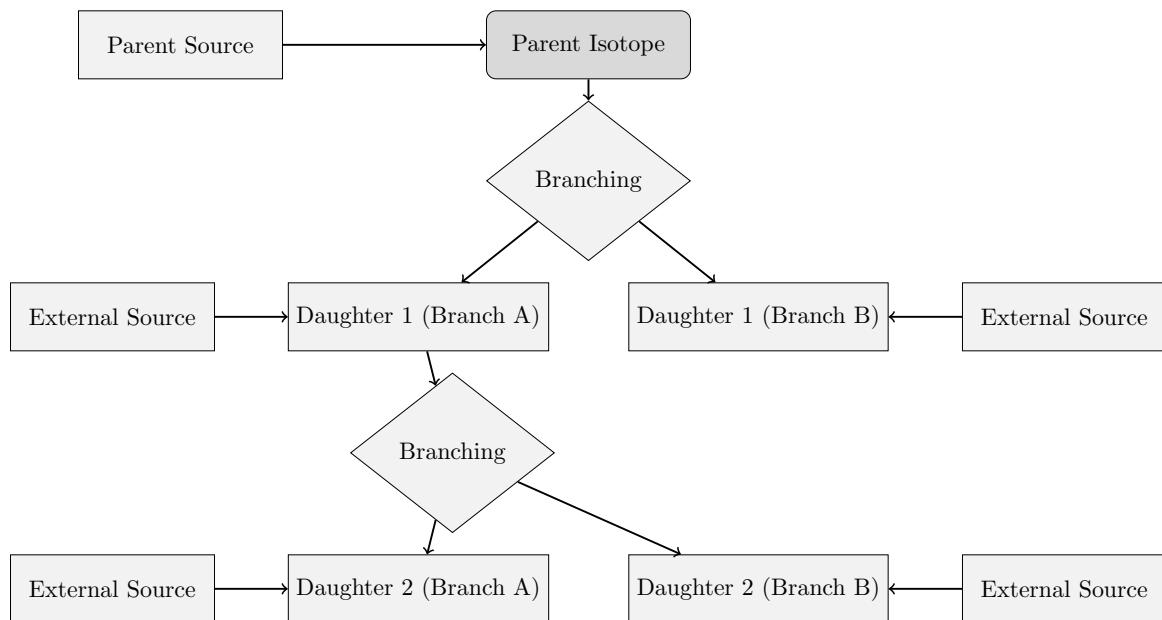


Figure 5.1: An example of several decay chains including branching factors and possible external source terms for each isotope on each chain.

#### 5.1.1 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.

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

### 5.1.2 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  $\lambda_i$
- $\vec{b}$  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  $i^{\text{th}}$  isotope
- $N_i(t)$  amount of the  $i^{\text{th}}$  isotope at time  $t$
- $N'_i(t)$  change in amount of the  $i^{\text{th}}$  isotope, with respect to time, at time  $t$

The differential equations for the parent isotope (first isotope), unstable daughter isotopes ( $i^{\text{th}}$  isotopes) and stable, final, daughter isotope (zth 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.

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

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

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

### 5.1.3 Numerical Inversion of the Laplace Transform

The Gaver-Stehfest[4] 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  $i^{\text{th}}$  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  $i^{\text{th}}$  isotope is reached (changing the equations appropriately for the parent, unstable daughter and stable daughter isotopes).

#### 5.1.4 Analytic Solution by Partial Fraction Expansion

The equation for the  $i^{\text{th}}$  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):

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

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  $m^{\text{th}}$  isotope in the decay chain, providing the  $m^{\text{th}}$  isotope is unstable.

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

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

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

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

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

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

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

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

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

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

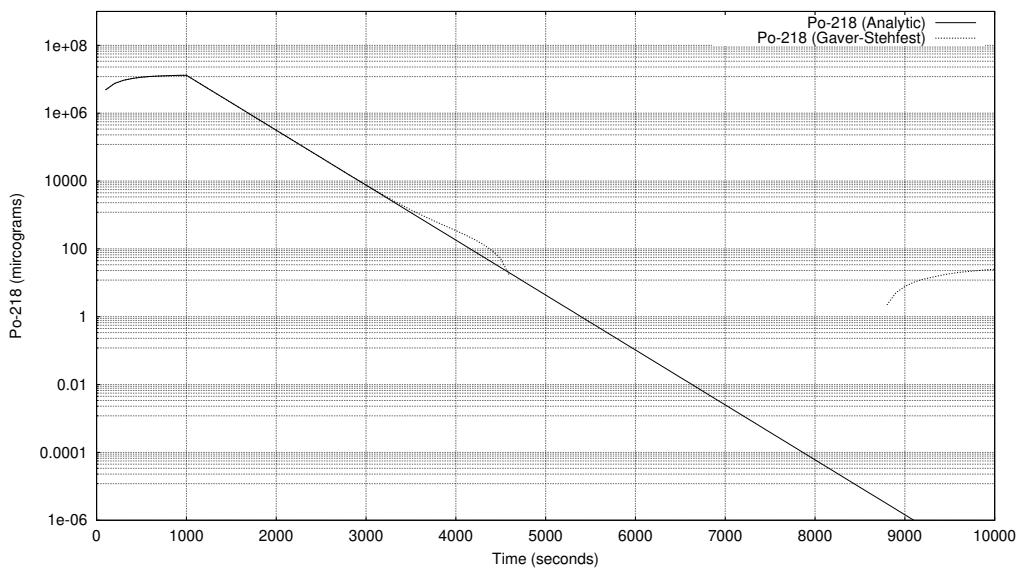


Figure 5.2: Decay of Po-218: Analytic and Gaver-Stehfest Calculations [jeff311]

## 5.2 Python Isotopes Class

The decay equations are computed within the isotopes class in the isotopes.py file.

```

1   class isotopes:
2
3   #####
4   # DECAY EQUATIONS
5   #####
6
7
8   @staticmethod
9   def calculate_activity(t, l, b, w, n0):
10  nt = numpy.zeros((len(n0),))
11  for m in range(0,len(n0)):
12      if(l[m] > 0.0):
13          nt[m] = isotopes.activity_unstable(t, l, b, w, n0, m)
14      elif(l[m] == 0.0):
15          nt[m] = isotopes.activity_stable(t, l, b, w, n0, m)
16  return nt
17
18 @staticmethod
19 def activity_unstable(t, l, b, w, n0, m):
20     s = 0.0
21     for k in range(0, m+1):
22         s = s + isotopes.r(k, m, b, l) * ( isotopes.f_unstable(t,k,m,l) * n0[k] + isotopes.g_unstable(t,
23                                         k,m,l) * w[k])
24
25 @staticmethod
26 def activity_stable(t, l, b, w, n0, m):
27     s = n0[m] + w[m] * t

```

```

28     for k in range(0, m):
29         s = s + isotopes.r(k, m, b, l) * (isotopes.f_stable(t,k,m,l) * n0[k] + isotopes.g_stable(t,k,m,l)
30             ) * w[k])
31     return s
32
33 @staticmethod
34 def r(k, m, b, l):
35     if(k == m):
36         return 1.0
37     else:
38         p = 1.0
39         for i in range(k, m):
40             p = p * (b[i] * l[i])
41         return p
42
43 @staticmethod
44 def f_unstable(t,k,m,l):
45     s = 0.0
46     for i in range(k, m+1):
47         p = 1.0
48         for j in range(k, m+1):
49             if(i != j):
50                 p = p * (1 / (l[i] - l[j]))
51             s = s + numpy.exp(-1 * l[i] * t) * p
52     s = (-1)**(m-k) * s
53     return s
54
55 @staticmethod
56 def g_unstable(t,k,m,l):
57     pa = 1.0
58     for i in range(k,m+1):
59         pa = pa * l[i]
60     pa = 1.0 / pa
61     s = 0.0
62     for i in range(k, m+1):
63         pb = 1.0
64         for j in range(k, m+1):
65             if(i != j):
66                 pb = pb * (1 / (l[i]-l[j]))
67             s = s + (1/l[i]) * numpy.exp(-l[i]*t) * pb
68     return pa + s * (-1)**(m-k+1)
69
70 @staticmethod
71 def f_stable(t,k,m_in,l):
72     m = m_in - 1
73
74     p = 1.0
75     for i in range(k, m+1):
76         p = p * l[i]
77
78     s = 0.0
79     for i in range(k, m+1):

```

```

80     r = l[i]
81     for j in range(k, m+1):
82         if(i != j):
83             r = r * (l[i] - l[j])
84         s = s + (1/r)*numpy.exp(-l[i]*t)
85
86     return (1.0/p) + s * (-1.0)**(m-k+1)
87
88
89 @staticmethod
90 def g_stable(t,k,m_in,l):
91     m = m_in - 1
92
93     pa = 1.0
94     for i in range(k,m+1):
95         pa = pa * l[i]
96     pa = 1.0 / pa
97
98     sa = 0.0
99     for i in range(k, m+1):
100        pb = 1.0
101        for j in range(k,m+1):
102            if(j != i):
103                pb = pb * l[j]
104            sa = sa + pb
105        pc = 1.0
106        for i in range(k, m+1):
107            pc = pc * l[i]**2
108
109        sb = 0.0
110        for i in range(k, m+1):
111            pd = 1.0
112            for j in range(k, m+1):
113                if(i != j):
114                    pd = pd * (1 / (l[i]-l[j]))
115            sb = sb + (1/(l[i]**2)) * numpy.exp(-l[i]*t) * pd
116
117    return pa * t + sa / pc + sb * (-1)**(m-k+1)

```

---

# Chapter 6

## Future Plans

### 6.1 Selection of Cross Section Databases

The Scanditronix MC40 Cyclotron at the University of Birmingham has several beamlines and is capable of accelerating protons, deuterons, Helium 3 and Helium 4 and fluxes and energy ranges detailed below.

Particle	Energy (MeV)	Max Current (micro A)	Flux (ions per second)
p	8-40	60	$3.75 \times 10^{14}$
d	8-40	30	$1.87 \times 10^{14}$
$^4He^{2+}$	8-53	30	$9.36 \times 10^{13}$
$^3He^{2+}$	4-20	60	$1.87 \times 10^{14}$

Table 6.1: Beam Characteristics of the Scanditronix MC-40

The current data file is for Protons, and the range of energies (for Fe-54 and Fe-56 at the very least) only range up to 30MeV. Previous versions of the TENDL data files have covered larger ranges (TENDL-2009 extended up to 200MeV).

As the energies do not cover the full range of possible energies for our own cyclotron, it would be desirable to use the TALYS program to calculate the reaction cross sections for a range up to at least 100MeV (for Protons, Deuterons, Helium 3 and Helium 4 ions) and save this cross section data into an alternate database.

# Bibliography

- [1] A.J. Koning et al. “TENDL: Complete Nuclear Data Library for Innovative Nuclear Science and Technology”. In: *Nuclear Data Sheets* 155 (2019).
- [2] A.J.M. Plompen et al. “The JEFF-3.3 Nuclear Data Library”. In: *Eur. Phys. J. A* 56 (181 2020).
- [3] 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.
- [4] H. Stehfest. “Algorithm 368: Numerical Inversion of Laplace Transform”. In: *Communications of the ACM* 13 (1970), pp. 47–49.

## Appendix D

# Activity V1 Full Results

### D.1 36MeV Proton Irradiated Iron

Fe irradiated by 36MeV protons: output file produced by Activity V1: <https://github.com/BenPalmer1983/activity>

Listing D.1: Isotope Activity at End of Simulation

---

```
1 =====
2 ACTIVITY code University of Birmingham
3 =====
4
5 Compiled: 15:08:41 11/01/2016
6 MPI Processes: 1
7
8 Date: 21:21 24/06/2016
9 Read user input          0.0000
10 Read Isotope Data       0.0000
11 Read Activity User Input File 0.0720
12 Read Trajectory File    0.0720
13 Data point count: 100887 0.9560
14
15 XS Data Files Loaded:
16 Reading xs file 1_1-26_54_0.dat
17 Reading xs file 1_1-26_56_0.dat
18 Reading xs file 1_1-26_57_0.dat
19 Reading xs file 1_1-26_58_0.dat
20
21 Calculation Settings
22
23 Beam Duration/s:      0.3000000000D+03
24 Beam Current/uA:       0.5000000000D+00
25 Beam Energy/MeV:       0.3600000000D+02
26 Beam Area/mm2:        0.1000000000D+03
27 Target Thickness/Angstrom: 0.5000000000D+07
28 Target Density/kgm-3:   0.8000000000D+04
29 Activity Measurement Time/s: 0.2600000000D+06
30 Input File:            /home/ben/activity/examples/Fe36MeV.in
31 Isotope File:          /home/ben/activity/data/isotopes.txt
32 Decay Modes File:     /home/ben/activity/data/decaymodes.txt
33 Gamma Energies File:  /home/ben/activity/data/gammaenergies.txt
34 XS File Directory:    /home/ben/activity/data/xs
35
36
37 Create material tally   1.2640
```

```

39 Fill tally with blank data           1.2640
40 Fill tally with isotope data       1.2640
41
42 -----
43 Starting Isotope Tally - Input Material
44 Element Z A M mk sim Half life     Decay Constant Activity      Reaction Rate Atoms/mg
45 -----
46   FE 026 054 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.5405383878D+28
47   FE 026 056 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.7890272999D+29
48   FE 026 057 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.1758807969D+28
49   FE 026 058 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.2263810953D+27
50 Total starting isotopes:        4
51
52
53 -----
54 Pre-Simulation Summary
55   1.2840
56 -----
57
58 Beam flux ions/s:            0.3120754681E+13
59
60
61 Calculate projectile-target-product reaction rates 1.2840
62 Reaction rates to calculate: 424
63 Trajectory fit equation:
64   0   0.3665497627E+05
65   1   -0.1783622595E-02
66   2   0.2947794093E-09
67   3   -0.3735372827E-16
68   4   0.1919778602E-23
69   5   -0.3566818320E-31
70 Affected material depth: 0.5000000000E+07
71
72 -----
73 Calculated variables
74 -----
75 Beam flux ions/s:            0.3120754681E+13
76 Affected depth/m:          0.5000000000E-03
77 Affected depth/ang:         0.5000000000E+07
78 Affected volume/m3:      0.500000054E-07
79 Number Density:            0.8626809746E+29
80 VPI:                      0.6020000000E+02
81 DPA:                      0.1306643605E-04
82
83 -----
84 Reaction rates in the target material 1.2840
85   Sym Z A M   Sym Z A M   Reaction Rate    Average XS    Number Density
86 -----
87   1:   FE 26 54 0 NN 0 1 0   0.3508066664E+09  0.4160433001E-28  0.5403805016E+28
88   2:   FE 26 54 0 H 1 1 0   0.1351444089E+10  0.1602761044E-27  0.5403805016E+28
89   3:   FE 26 54 0 H 1 2 0   0.3205323639E+08  0.3801391343E-29  0.5403805016E+28
90   4:   FE 26 54 0 H 1 3 0   0.7165361660E+06  0.8497845101E-31  0.5403805016E+28
91   5:   FE 26 54 0 HE 2 3 0   0.8617826538E+07  0.1022041294E-29  0.5403805016E+28
92   6:   FE 26 54 0 HE 2 4 0   0.5675883691E+08  0.6731381151E-29  0.5403805016E+28
93   67:  FE 26 54 0 TI 22 46 0   0.4687831440E+01  0.5559588940E-36  0.5403805016E+28
94   76:  FE 26 54 0 V 23 47 0   0.3452942314E+06  0.4095057628E-31  0.5403805016E+28
95   78:  FE 26 54 0 V 23 49 0   0.5791127831E+07  0.6868056297E-30  0.5403805016E+28
96   85:  FE 26 54 0 CR 24 49 0   0.2416421986E+07  0.2865784131E-30  0.5403805016E+28
97   86:  FE 26 54 0 CR 24 50 0   0.4290008294E+08  0.5087785892E-29  0.5403805016E+28

```

98	87:	FE	26	54	0	CR	24	51	0	0.2135062196E+07	0.2532102172E-30	0.5403805016E+28
99	88:	FE	26	54	0	CR	24	52	0	0.8659953579E+08	0.1027037400E-28	0.5403805016E+28
100	92:	FE	26	54	0	MN	25	50	0	0.1542445948E+07	0.1829281950E-30	0.5403805016E+28
101	93:	FE	26	54	0	MN	25	51	0	0.1778372502E+07	0.2109081827E-30	0.5403805016E+28
102	94:	FE	26	54	0	MN	25	52	0	0.9863724488E+08	0.1169800030E-28	0.5403805016E+28
103	95:	FE	26	54	0	MN	25	53	0	0.2232277074E+09	0.2647395303E-28	0.5403805016E+28
104	99:	FE	26	54	0	FE	26	52	0	0.1143967233E+08	0.1356701421E-29	0.5403805016E+28
105	100:	FE	26	54	0	FE	26	53	0	0.1888610876E+09	0.2239820325E-28	0.5403805016E+28
106	101:	FE	26	54	0	FE	26	54	0	0.1103802931E+09	0.1309068094E-28	0.5403805016E+28
107	104:	FE	26	54	0	CO	27	53	0	0.4251641576E+06	0.5042284431E-31	0.5403805016E+28
108	105:	FE	26	54	0	CO	27	54	0	0.8026455518E+07	0.9519069510E-30	0.5403805016E+28
109	106:	FE	26	54	0	CO	27	55	0	0.1175568630E+06	0.1394179471E-31	0.5403805016E+28
110	107:	FE	26	56	0	NN	0	1	0	0.1294835964E+11	0.1052010224E-27	0.7887968323E+29
111	108:	FE	26	56	0	H	1	1	0	0.1322708565E+11	0.1074655765E-27	0.7887968323E+29
112	109:	FE	26	56	0	H	1	2	0	0.6485018658E+09	0.5268857307E-29	0.7887968323E+29
113	110:	FE	26	56	0	H	1	3	0	0.2804930290E+08	0.2278910553E-30	0.7887968323E+29
114	111:	FE	26	56	0	HE	2	3	0	0.1152252073E+09	0.93616556570E-30	0.7887968323E+29
115	112:	FE	26	56	0	HE	2	4	0	0.8396082900E+09	0.6821532069E-29	0.7887968323E+29
116	169:	FE	26	56	0	TI	22	48	0	0.7417367941E+02	0.6026359420E-36	0.7887968323E+29
117	179:	FE	26	56	0	V	23	49	0	0.1002075285E+08	0.8141521197E-31	0.7887968323E+29
118	181:	FE	26	56	0	V	23	51	0	0.3129496674E+07	0.2542609711E-31	0.7887968323E+29
119	188:	FE	26	56	0	CR	24	50	0	0.3047989029E+03	0.2476387520E-35	0.7887968323E+29
120	189:	FE	26	56	0	CR	24	51	0	0.1903672236E+09	0.1546669008E-29	0.7887968323E+29
121	190:	FE	26	56	0	CR	24	52	0	0.2832038545E+09	0.2300935089E-29	0.7887968323E+29
122	191:	FE	26	56	0	CR	24	53	0	0.1158364999E+07	0.9411322022E-32	0.7887968323E+29
123	192:	FE	26	56	0	CR	24	54	0	0.1165756880E+08	0.9471378543E-31	0.7887968323E+29
124	196:	FE	26	56	0	MN	25	51	0	0.9817102039E+01	0.7976061834E-37	0.7887968323E+29
125	197:	FE	26	56	0	MN	25	52	0	0.1545890268E+09	0.1255983315E-29	0.7887968323E+29
126	198:	FE	26	56	0	MN	25	53	0	0.8815593471E+08	0.7162370153E-30	0.7887968323E+29
127	199:	FE	26	56	0	MN	25	54	0	0.1267995071E+10	0.1030202911E-28	0.7887968323E+29
128	200:	FE	26	56	0	MN	25	55	0	0.7253812202E+09	0.5893475939E-29	0.7887968323E+29
129	204:	FE	26	56	0	FE	26	53	0	0.2156782879E+02	0.1752312805E-36	0.7887968323E+29
130	205:	FE	26	56	0	FE	26	54	0	0.2975937641E+10	0.2417848215E-28	0.7887968323E+29
131	206:	FE	26	56	0	FE	26	55	0	0.4948211136E+10	0.4020253414E-28	0.7887968323E+29
132	207:	FE	26	56	0	FE	26	56	0	0.1551234326E+10	0.1260325181E-28	0.7887968323E+29
133	210:	FE	26	56	0	CO	27	55	0	0.3083905589E+09	0.2505568504E-29	0.7887968323E+29
134	211:	FE	26	56	0	CO	27	56	0	0.2971765420E+09	0.2414458427E-29	0.7887968323E+29
135	212:	FE	26	56	0	CO	27	57	0	0.3041404939E+07	0.2471038170E-31	0.7887968323E+29
136	213:	FE	26	57	0	NN	0	1	0	0.3649443293E+09	0.1330164742E-27	0.1758294237E+28
137	214:	FE	26	57	0	H	1	1	0	0.2670335825E+09	0.9732954535E-28	0.1758294237E+28
138	215:	FE	26	57	0	H	1	2	0	0.1672940269E+08	0.6097604437E-29	0.1758294237E+28
139	216:	FE	26	57	0	H	1	3	0	0.1275440383E+07	0.4648779807E-30	0.1758294237E+28
140	217:	FE	26	57	0	HE	2	3	0	0.2836366558E+07	0.1033811047E-29	0.1758294237E+28
141	218:	FE	26	57	0	HE	2	4	0	0.2332170710E+08	0.8500395818E-29	0.1758294237E+28
142	273:	FE	26	57	0	TI	22	49	0	0.7752833017E+00	0.2825785826E-36	0.1758294237E+28
143	282:	FE	26	57	0	V	23	49	0	0.3281858226E+02	0.1196185760E-34	0.1758294237E+28
144	283:	FE	26	57	0	V	23	50	0	0.1835618293E+06	0.6690540276E-31	0.1758294237E+28
145	284:	FE	26	57	0	V	23	51	0	0.6679861824E+00	0.2434704685E-36	0.1758294237E+28
146	285:	FE	26	57	0	V	23	52	0	0.3050010636E+01	0.1111680957E-35	0.1758294237E+28
147	293:	FE	26	57	0	CR	24	51	0	0.3269383508E+04	0.1191638922E-32	0.1758294237E+28
148	294:	FE	26	57	0	CR	24	52	0	0.7230173808E+07	0.2635284756E-29	0.1758294237E+28
149	295:	FE	26	57	0	CR	24	53	0	0.1653080211E+07	0.6025217646E-30	0.1758294237E+28
150	296:	FE	26	57	0	CR	24	54	0	0.4518951495E+05	0.1647086821E-31	0.1758294237E+28
151	297:	FE	26	57	0	CR	24	55	0	0.7219801309E+04	0.2631504144E-32	0.1758294237E+28
152	301:	FE	26	57	0	MN	25	52	0	0.2058316554E+07	0.7502240448E-30	0.1758294237E+28
153	302:	FE	26	57	0	MN	25	53	0	0.9698739328E+07	0.3535038104E-29	0.1758294237E+28
154	303:	FE	26	57	0	MN	25	54	0	0.2412989059E+07	0.8794966005E-30	0.1758294237E+28
155	304:	FE	26	57	0	MN	25	55	0	0.1977898329E+08	0.7209128654E-29	0.1758294237E+28
156	305:	FE	26	57	0	MN	25	56	0	0.5440385638E+07	0.1982935089E-29	0.1758294237E+28
157	309:	FE	26	57	0	FE	26	54	0	0.2296404190E+07	0.8370032473E-30	0.1758294237E+28
158	310:	FE	26	57	0	FE	26	55	0	0.9449535161E+08	0.3444207100E-28	0.1758294237E+28

159 311: FE 26 57 0 FE 26 56 0 0.1023020929E+09 0.3728750553E-28 0.1758294237E+28  
 160 312: FE 26 57 0 FE 26 57 0 0.2808424683E+08 0.1023626672E-28 0.1758294237E+28  
 161 315: FE 26 57 0 CO 27 55 0 0.4515569494E+07 0.1645854134E-29 0.1758294237E+28  
 162 316: FE 26 57 0 CO 27 56 0 0.1114292806E+08 0.4061422208E-29 0.1758294237E+28  
 163 317: FE 26 57 0 CO 27 57 0 0.5878105202E+07 0.2142476992E-29 0.1758294237E+28  
 164 318: FE 26 57 0 CO 27 58 0 0.3965181061E+05 0.1445246198E-31 0.1758294237E+28  
 165 319: FE 26 58 0 NN 0 1 0 0.5793566623E+08 0.1640601429E-27 0.2263149715E+27  
 166 320: FE 26 58 0 H 1 1 0 0.2791132972E+08 0.7903830301E-28 0.2263149715E+27  
 167 321: FE 26 58 0 H 1 2 0 0.2037087602E+07 0.5768551650E-29 0.2263149715E+27  
 168 322: FE 26 58 0 H 1 3 0 0.1593981581E+06 0.4513779906E-30 0.2263149715E+27  
 169 323: FE 26 58 0 HE 2 3 0 0.2598326212E+06 0.7357847030E-30 0.2263149715E+27  
 170 324: FE 26 58 0 HE 2 4 0 0.2551528049E+07 0.7225325669E-29 0.2263149715E+27  
 171 376: FE 26 58 0 TI 22 50 0 0.9741805634E+00 0.2758649600E-35 0.2263149715E+27  
 172 385: FE 26 58 0 V 23 50 0 0.2503943210E+00 0.7090576626E-36 0.2263149715E+27  
 173 386: FE 26 58 0 V 23 51 0 0.9310532844E+04 0.2636523317E-31 0.2263149715E+27  
 174 388: FE 26 58 0 V 23 53 0 0.1852730029E-06 0.5246494484E-42 0.2263149715E+27  
 175 396: FE 26 58 0 CR 24 52 0 0.4688542450E+04 0.1327684645E-31 0.2263149715E+27  
 176 397: FE 26 58 0 CR 24 53 0 0.1090733133E+06 0.3088698988E-30 0.2263149715E+27  
 177 398: FE 26 58 0 CR 24 54 0 0.7904949881E+05 0.2238495372E-30 0.2263149715E+27  
 178 399: FE 26 58 0 CR 24 55 0 0.5894353392E-05 0.1669141865E-40 0.2263149715E+27  
 179 400: FE 26 58 0 CR 24 56 0 0.3341354293E-04 0.9461927315E-40 0.2263149715E+27  
 180 405: FE 26 58 0 MN 25 53 0 0.1200142917E+07 0.3398521693E-29 0.2263149715E+27  
 181 406: FE 26 58 0 MN 25 54 0 0.1053914972E+07 0.2984438639E-29 0.2263149715E+27  
 182 407: FE 26 58 0 MN 25 55 0 0.2219588486E+06 0.6285351112E-30 0.2263149715E+27  
 183 408: FE 26 58 0 MN 25 56 0 0.4947583955E+06 0.1401039090E-29 0.2263149715E+27  
 184 409: FE 26 58 0 MN 25 57 0 0.3707982841E+06 0.1050013290E-29 0.2263149715E+27  
 185 414: FE 26 58 0 FE 26 55 0 0.4036879251E+06 0.1143148996E-29 0.2263149715E+27  
 186 415: FE 26 58 0 FE 26 56 0 0.1504003971E+08 0.4258984535E-28 0.2263149715E+27  
 187 416: FE 26 58 0 FE 26 57 0 0.8926880873E+07 0.2527882127E-28 0.2263149715E+27  
 188 417: FE 26 58 0 FE 26 58 0 0.3876895164E+07 0.1097845276E-28 0.2263149715E+27  
 189 421: FE 26 58 0 CO 27 56 0 0.2685676346E+07 0.7605202007E-29 0.2263149715E+27  
 190 422: FE 26 58 0 CO 27 57 0 0.3377108415E+07 0.9563174557E-29 0.2263149715E+27  
 191 423: FE 26 58 0 CO 27 58 0 0.7015543919E+06 0.1986636580E-29 0.2263149715E+27  
 192 424: FE 26 58 0 CO 27 59 0 0.7735041844E+04 0.2190381423E-31 0.2263149715E+27  
 193  
 194  
 195 Prepare full isotope tally 1.2840  
 196  
 197 -----

198 Full Starting Isotope Tally  
 199 Key Element Z A M mk sim Half life Activity Reaction Rate Atoms  
 200 -----

201	000001	NN	000	001	0	000	001	-0.1000000000D+01	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
202	000591	H	001	001	0	000	001	-0.1000000000D+01	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
203	000592	H	001	002	0	000	001	-0.1000000000D+01	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
204	000593	H	001	003	0	000	001	0.3885235200D+09	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
205	001183	HE	002	003	0	000	001	-0.1000000000D+01	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
206	001184	HE	002	004	0	000	001	-0.1000000000D+01	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
207	008881	P	015	031	0	000	001	-0.1000000000D+01	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
208	009472	S	016	032	0	000	001	-0.1000000000D+01	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
209	009473	S	016	033	0	000	001	-0.1000000000D+01	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
210	009474	S	016	034	0	000	001	-0.1000000000D+01	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
211	009476	S	016	036	0	000	001	-0.1000000000D+01	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
212	010062	CL	017	032	0	000	001	0.2980000000D+00	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
213	010063	CL	017	033	0	000	001	0.2511000000D+01	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
214	010064	CL	017	034	0	000	001	0.1526400000D+01	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
215	010065	CL	017	035	0	000	001	-0.1000000000D+01	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
216	010066	CL	017	036	0	000	001	0.9492336000D+13	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00
217	010067	CL	017	037	0	000	001	-0.1000000000D+01	0.0000000000D+00	0.0000000000D+00	0.0000000000D+00

218 010068 CL 017 038 0 000 001 0.2234400000D+04 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 219 010069 CL 017 039 0 000 001 0.3372000000D+04 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 220 010070 CL 017 040 0 000 001 0.8100000000D+02 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 221 010071 CL 017 041 0 000 001 0.3840000000D+02 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 222 010653 AR 018 033 0 000 001 0.1730000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 223 010654 AR 018 034 0 000 001 0.8445000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 224 010655 AR 018 035 0 000 001 0.1775000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 225 010656 AR 018 036 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 226 010657 AR 018 037 0 000 001 0.3027456000D+07 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 227 010658 AR 018 038 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 228 010659 AR 018 039 0 000 001 0.8483184000D+10 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 229 010660 AR 018 040 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 230 010661 AR 018 041 0 000 001 0.6576600000D+04 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 231 010662 AR 018 042 0 000 001 0.1037534400D+10 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 232 010663 AR 018 043 0 000 001 0.3222000000D+03 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 233 011245 K 019 035 0 000 001 0.1780000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 234 011246 K 019 036 0 000 001 0.3420000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 235 011247 K 019 037 0 000 001 0.1226000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 236 011248 K 019 038 0 000 001 0.4581600000D+03 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 237 011249 K 019 039 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 238 011250 K 019 040 0 000 001 0.3935692800D+17 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 239 011251 K 019 041 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 240 011252 K 019 042 0 000 001 0.4449600000D+05 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 241 011253 K 019 043 0 000 001 0.8028000000D+05 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 242 011254 K 019 044 0 000 001 0.1327800000D+04 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 243 011255 K 019 045 0 000 001 0.1038000000D+04 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 244 011256 K 019 046 0 000 001 0.1050000000D+03 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 245 011837 CA 020 037 0 000 001 0.1811000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 246 011838 CA 020 038 0 000 001 0.4400000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 247 011839 CA 020 039 0 000 001 0.8596000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 248 011840 CA 020 040 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 249 011841 CA 020 041 0 000 001 0.3216672000D+13 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 250 011842 CA 020 042 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 251 011843 CA 020 043 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 252 011844 CA 020 044 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 253 011845 CA 020 045 0 000 001 0.1404950400D+08 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 254 011846 CA 020 046 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 255 011847 CA 020 047 0 000 001 0.3919104000D+06 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 256 011848 CA 020 048 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 257 012429 SC 021 039 0 000 001 0.3000000000D-06 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 258 012430 SC 021 040 0 000 001 0.1823000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 259 012431 SC 021 041 0 000 001 0.5963000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 260 012432 SC 021 042 0 000 001 0.6813000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 261 012433 SC 021 043 0 000 001 0.1400760000D+05 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 262 012434 SC 021 044 0 000 001 0.1429200000D+05 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 263 012435 SC 021 045 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 264 012436 SC 021 046 0 000 001 0.7239456000D+07 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 265 012437 SC 021 047 0 000 001 0.2893708800D+06 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 266 012438 SC 021 048 0 000 001 0.1572120000D+06 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 267 012439 SC 021 049 0 000 001 0.3432000000D+04 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 268 012440 SC 021 050 0 000 001 0.1025000000D+03 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 269 013021 TI 022 041 0 000 001 0.8040000000D-01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 270 013022 TI 022 042 0 000 001 0.1990000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 271 013023 TI 022 043 0 000 001 0.5090000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 272 013024 TI 022 044 0 000 001 0.1892160000D+10 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 273 013025 TI 022 045 0 000 001 0.1108800000D+05 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 274 013026 TI 022 046 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 275 013027 TI 022 047 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 276 013028 TI 022 048 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 277 013029 TI 022 049 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 278 013030 TI 022 050 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00

279 013031 TI 022 051 0 000 001 0.3456000000D+03 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 280 013032 TI 022 052 0 000 001 0.1020000000D+03 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 281 013613 V 023 043 0 000 001 0.8000000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 282 013614 V 023 044 0 000 001 0.1110000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 283 013615 V 023 045 0 000 001 0.5470000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 284 013616 V 023 046 0 000 001 0.4225000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 285 013617 V 023 047 0 000 001 0.1956000000D+04 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 286 013618 V 023 048 0 000 001 0.1380110400D+07 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 287 013619 V 023 049 0 000 001 0.2851200000D+08 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 288 013620 V 023 050 0 000 001 0.4415040000D+25 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 289 013621 V 023 051 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 290 013622 V 023 052 0 000 001 0.2245800000D+03 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 291 013623 V 023 053 0 000 001 0.9600000000D+02 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 292 013624 V 023 054 0 000 001 0.4980000000D+02 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 293 013625 V 023 055 0 000 001 0.6540000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 294 014205 CR 024 045 0 000 001 0.5000000000D-01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 295 014206 CR 024 046 0 000 001 0.2600000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 296 014207 CR 024 047 0 000 001 0.5000000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 297 014208 CR 024 048 0 000 001 0.7776160000D+05 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 298 014209 CR 024 049 0 000 001 0.2538000000D+04 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 299 014210 CR 024 050 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 300 014211 CR 024 051 0 000 001 0.2393496000D+07 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 301 014212 CR 024 052 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 302 014213 CR 024 053 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 303 014214 CR 024 054 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 304 014215 CR 024 055 0 000 001 0.2098200000D+03 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 305 014216 CR 024 056 0 000 001 0.3564000000D+03 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 306 014797 MN 025 047 0 000 001 0.1000000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 307 014798 MN 025 048 0 000 001 0.1581000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 308 014799 MN 025 049 0 000 001 0.3820000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 309 014800 MN 025 050 0 000 001 0.2838800000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 310 014801 MN 025 051 0 000 001 0.2772000000D+04 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 311 014802 MN 025 052 0 000 001 0.4830624000D+06 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 312 014803 MN 025 053 0 000 001 0.1179446400D+15 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 313 014804 MN 025 054 0 000 001 0.2696716800D+08 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 314 014805 MN 025 055 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 315 014806 MN 025 056 0 000 001 0.9284040000D+04 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 316 014807 MN 025 057 0 000 001 0.8540000000D+02 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 317 015389 FE 026 049 0 000 001 0.7000000000D-01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 318 015390 FE 026 050 0 000 001 0.1550000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 319 015391 FE 026 051 0 000 001 0.3050000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 320 015392 FE 026 052 0 000 001 0.2979000000D+05 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 321 015393 FE 026 053 0 000 001 0.5106000000D+03 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 322 015394 FE 026 054 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.5405383878D+28  
 323 015395 FE 026 055 0 000 001 0.8631403200D+08 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 324 015396 FE 026 056 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.7890272999D+29  
 325 015397 FE 026 057 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.1758807969D+28  
 326 015398 FE 026 058 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.2263810953D+27  
 327 015981 CO 027 051 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 328 015982 CO 027 052 0 000 001 0.1150000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 329 015983 CO 027 053 0 000 001 0.2400000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 330 015984 CO 027 054 0 000 001 0.1932300000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 331 015985 CO 027 055 0 000 001 0.6310800000D+05 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 332 015986 CO 027 056 0 000 001 0.6672931200D+07 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 333 015987 CO 027 057 0 000 001 0.2347833600D+08 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 334 015988 CO 027 058 0 000 001 0.6122304000D+07 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 335 015989 CO 027 059 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
 336  
 337  
 338  
 339

```

340
341
342
343
344
345
346
347 =====

348
349 START of SIMULATION
350
351 Simulation time from 0.000000000E+00 to 0.260000000E+06
352 Beam from 0.000000000E+00 to 0.300000000E+03
353
354 =====

355
356
357
358
359 -----
360 Simulation tally at 0.300000000E+03
361 -----
362
363 ----

364 Isotope Tally 1.2880
365 Key Element Z A M mk sim Half life Decay Constant Activity Reaction Rate Start Atoms Beam
      End Atoms Atoms At Time
366 ----

367 000001 NN 000 001 0 000 001 -0.100000000D+01 0.000000000D+00 0.000000000D+00 0.000000000D+00 0.000000000D
      +00 0.4116613889D+13 0.4116613889D+13
368 000593 H 001 003 0 000 001 0.3885235200D+09 0.1784054619D-08 0.1616389319D+02 0.000000000D+00 0.000000000D
      +00 0.9060200854D+10 0.9060200854D+10
369 001183 HE 002 003 0 000 001 -0.100000000D+01 0.000000000D+00 0.000000000D+00 0.000000000D+00 0.000000000D
      +00 0.242400000D+04 0.242400000D+04
370 013027 TI 022 047 0 000 001 -0.100000000D+01 0.000000000D+00 0.000000000D+00 0.000000000D+00 0.000000000D
      +00 0.5316233671D+07 0.5316233671D+07
371 013029 TI 022 049 0 000 001 -0.100000000D+01 0.000000000D+00 0.000000000D+00 0.000000000D+00 0.000000000D
      +00 0.1736864062D+05 0.1736864062D+05
372 013617 V 023 047 0 000 001 0.195600000D+04 0.3543697242D-03 0.3482463420D+05 0.000000000D+00 0.000000000D
      +00 0.9827203574D+08 0.9827203574D+08
373 013619 V 023 049 0 000 001 0.285120000D+08 0.2431071761D-07 0.1160219131D+03 0.000000000D+00 0.000000000D
      +00 0.4772459413D+10 0.4772459413D+10
374 013621 V 023 051 0 000 001 -0.100000000D+01 0.000000000D+00 0.000000000D+00 0.000000000D+00 0.000000000D
      +00 0.2509195042D+07 0.2509195042D+07
375 013622 V 023 052 0 000 001 0.224580000D+03 0.3086415445D-02 0.1841703348D+01 0.000000000D+00 0.000000000D
      +00 0.5967127177D+03 0.5967127177D+03
376 013623 V 023 053 0 000 001 0.960000000D+02 0.7220283131D-02 0.1640359913D-06 0.000000000D+00 0.000000000D
      +00 0.2271877547D-04 0.2271877547D-04
377 014209 CR 024 049 0 000 001 0.253800000D+04 0.2731076362D-03 0.1900894319D+06 0.000000000D+00 0.000000000D
      +00 0.6960238630D+09 0.6960238630D+09
378 014210 CR 024 050 0 000 001 -0.100000000D+01 0.000000000D+00 0.000000000D+00 0.000000000D+00 0.000000000D
      +00 0.4621020723D+09 0.4621020723D+09
379 014211 CR 024 051 0 000 001 0.2393496000D+07 0.2895961307D-06 0.1672958542D+05 0.000000000D+00 0.000000000D
      +00 0.5776867728D+11 0.5776867728D+11
380 014212 CR 024 052 0 000 001 -0.100000000D+01 0.000000000D+00 0.000000000D+00 0.000000000D+00 0.000000000D
      +00 0.1648354473D+08 0.1648354473D+08
381 014213 CR 024 053 0 000 001 -0.100000000D+01 0.000000000D+00 0.000000000D+00 0.000000000D+00 0.000000000D

```

+00 0.4169728000D+07 0.4169728000D+07  
 382 014214 CR 024 054 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
     +00 0.1470632000D+07 0.1470632000D+07  
 383 014215 CR 024 055 0 000 001 0.2098200000D+03 0.3303532459D-02 0.4539932896D+04 0.0000000000D+00 0.0000000000D+00  
     +00 0.1374266169D+07 0.1374266169D+07  
 384 014216 CR 024 056 0 000 001 0.3564000000D+03 0.1944857409D-02 0.1476992152D-04 0.0000000000D+00 0.0000000000D+00  
     +00 0.7594346737D-02 0.7594346737D-02  
 385 014800 MN 025 050 0 000 001 0.2838800000D+00 0.2441690787D+01 0.1542445948D+07 0.0000000000D+00 0.0000000000D+00  
     +00 0.6317122367D+06 0.6317122367D+06  
 386 014801 MN 025 051 0 000 001 0.2772000000D+04 0.2500530954D-03 0.1285259866D+06 0.0000000000D+00 0.0000000000D+00  
     +00 0.5139947831D+09 0.5139947831D+09  
 387 014802 MN 025 052 0 000 001 0.4830624000D+06 0.1434901952D-05 0.1098860016D+06 0.0000000000D+00 0.0000000000D+00  
     +00 0.7658084333D+11 0.7658084333D+11  
 388 014803 MN 025 053 0 000 001 0.1179446400D+15 0.5876885805D-14 0.6277577829D-03 0.0000000000D+00 0.0000000000D+00  
     +00 0.1068181012D+12 0.1068181012D+12  
 389 014804 MN 025 054 0 000 001 0.2696716800D+08 0.2570337310D-07 0.9804220660D+04 0.0000000000D+00 0.0000000000D+00  
     +00 0.3814371220D+12 0.3814371220D+12  
 390 014805 MN 025 055 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00  
     +00 0.2614130226D+07 0.2614130226D+07  
 391 014806 MN 025 056 0 000 001 0.9284040000D+04 0.7466008123D-04 0.1314578074D+06 0.0000000000D+00 0.0000000000D+00  
     +00 0.1760750930D+10 0.1760750930D+10  
 392 014807 MN 025 057 0 000 001 0.8540000000D+02 0.8116477524D-02 0.3383153467D+06 0.0000000000D+00 0.0000000000D+00  
     +00 0.4168253355D+08 0.4168253355D+08  
 393 015392 FE 026 052 0 000 001 0.2979000000D+05 0.2326778048D-04 0.7957468329D+05 0.0000000000D+00 0.0000000000D+00  
     +00 0.3419951609D+10 0.3419951609D+10  
 394 015393 FE 026 053 0 000 001 0.5106000000D+03 0.1357515042D-02 0.6332089595D+08 0.0000000000D+00 0.0000000000D+00  
     +00 0.4664471036D+11 0.4664471036D+11  
 395 015394 FE 026 054 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.5405383878D+28  
     +28 0.5405383878D+28 0.5405383878D+28  
 396 015395 FE 026 055 0 000 001 0.8631403200D+08 0.8030527187D-08 0.1215087645D+05 0.0000000000D+00 0.0000000000D+00  
     +00 0.1513085775D+13 0.1513085775D+13  
 397 015396 FE 026 056 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.7890272999D+29  
     +29 0.7890272999D+29 0.7890272999D+29  
 398 015397 FE 026 057 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.1758807969D+28  
     +28 0.1758807969D+28 0.1758807969D+28  
 399 015398 FE 026 058 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.2263810953D+27  
     +27 0.2263810953D+27 0.2263810953D+27  
 400 015983 CO 027 053 0 000 001 0.2400000000D+00 0.2888113252D+01 0.4251641576D+06 0.0000000000D+00 0.0000000000D+00  
     +00 0.1472117332D+06 0.1472117332D+06  
 401 015984 CO 027 054 0 000 001 0.1932300000D+00 0.3587161313D+01 0.8026455518D+07 0.0000000000D+00 0.0000000000D+00  
     +00 0.2237550759D+07 0.2237550759D+07  
 402 015985 CO 027 055 0 000 001 0.6310800000D+05 0.1098350733D-04 0.1029731940D+07 0.0000000000D+00 0.0000000000D+00  
     +00 0.9375256095D+11 0.9375256095D+11  
 403 015986 CO 027 056 0 000 001 0.6672931200D+07 0.1038744683D-06 0.9691497262D+04 0.0000000000D+00 0.0000000000D+00  
     +00 0.9330009018D+11 0.9330009018D+11  
 404 015987 CO 027 057 0 000 001 0.2347833600D+08 0.2952284100D-07 0.1089088520D+03 0.0000000000D+00 0.0000000000D+00  
     +00 0.3688969230D+10 0.3688969230D+10  
 405 015988 CO 027 058 0 000 001 0.6122304000D+07 0.1132167205D-06 0.2517465310D+02 0.0000000000D+00 0.0000000000D+00  
     +00 0.2223580845D+09 0.2223580845D+09  
 406  
 407 -----  
 408 Simulation tally at 0.2600000000E+06  
 409 -----  
 410  
 411 -----  
 412 Isotope Tally 1.7640  
 413 Key Element Z A M mk sim Half life Decay Constant Activity Reaction Rate Start Atoms Beam  
     End Atoms Atoms At Time  
 414 -----

415 000001 NN 000 001 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D  
+00 0.4116613889D+13 0.4116613889D+13

416 000593 H 001 003 0 000 001 0.3885235200D+09 0.1784054619D-08 0.1615640588D+02 0.0000000000D+00 0.0000000000D  
+00 0.9060200854D+10 0.9056004063D+10

417 001183 HE 002 003 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D  
+00 0.2424000000D+04 0.4199214759D+07

418 013027 TI 022 047 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D  
+00 0.5316233671D+07 0.1035882694D+09

419 013029 TI 022 049 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D  
+00 0.1736364062D+05 0.6440824599D+08

420 013619 V 023 049 0 000 001 0.2851200000D+08 0.2431071761D-07 0.1313773643D+03 0.0000000000D+00 0.0000000000D  
+00 0.4772459413D+10 0.5404092399D+10

421 013621 V 023 051 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D  
+00 0.2509195042D+07 0.8409826886D+10

422 014210 CR 024 050 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D  
+00 0.4621020723D+09 0.4627337845D+09

423 014211 CR 024 051 0 000 001 0.2393496000D+07 0.2895961307D-06 0.1444370965D+05 0.0000000000D+00 0.0000000000D  
+00 0.5776867728D+11 0.4987535437D+11

424 014212 CR 024 052 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D  
+00 0.1648354473D+08 0.4857309090D+11

425 014213 CR 024 053 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D  
+00 0.4169728000D+07 0.4170359063D+07

426 014214 CR 024 054 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D  
+00 0.1470632000D+07 0.2539147607D+10

427 014802 MN 025 052 0 000 001 0.4830624000D+06 0.1434901952D-05 0.4510767081D+05 0.0000000000D+00 0.0000000000D  
+00 0.7658084333D+11 0.3143606485D+11

428 014803 MN 025 053 0 000 001 0.1179446400D+15 0.5876885805D-14 0.1176009917D-02 0.0000000000D+00 0.0000000000D  
+00 0.1068181012D+12 0.2001076685D+12

429 014804 MN 025 054 0 000 001 0.2696716800D+08 0.2570337310D-07 0.9738993802D+04 0.0000000000D+00 0.0000000000D  
+00 0.3814371220D+12 0.3788994450D+12

430 014805 MN 025 055 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.0000000000D  
+00 0.2614130226D+07 0.6439401457D+10

431 015392 FE 026 052 0 000 001 0.2979000000D+05 0.2326778048D-04 0.1890116552D+03 0.0000000000D+00 0.0000000000D  
+00 0.3419951609D+10 0.8123321232D+07

432 015394 FE 026 054 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.5405383878D  
+28 0.5405383878D+28 0.5405383878D+28

433 015395 FE 026 055 0 000 001 0.8631403200D+08 0.8030527187D-08 0.1280863433D+05 0.0000000000D+00 0.0000000000D  
+00 0.1513085775D+13 0.1594992960D+13

434 015396 FE 026 056 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.7890272999D  
+29 0.7890272999D+29 0.7890272999D+29

435 015397 FE 026 057 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.1758807969D  
+28 0.1758807969D+28 0.1758807969D+28

436 015398 FE 026 058 0 000 001 -0.1000000000D+01 0.0000000000D+00 0.0000000000D+00 0.0000000000D+00 0.2263810953D  
+27 0.2263810953D+27 0.2263810953D+27

437 015985 CO 027 055 0 000 001 0.6310800000D+05 0.1098350733D-04 0.5942036078D+05 0.0000000000D+00 0.0000000000D  
+00 0.9375256095D+11 0.5409962319D+10

438 015986 CO 027 056 0 000 001 0.6672931200D+07 0.1038744683D-06 0.9433552328D+04 0.0000000000D+00 0.0000000000D  
+00 0.9330009018D+11 0.9081685308D+11

439 015987 CO 027 057 0 000 001 0.2347833600D+08 0.2952284100D-07 0.1080770318D+03 0.0000000000D+00 0.0000000000D  
+00 0.3688969230D+10 0.3660793752D+10

440 015988 CO 027 058 0 000 001 0.6122304000D+07 0.1132167205D-06 0.2444523420D+02 0.0000000000D+00 0.0000000000D  
+00 0.2223580845D+09 0.2159154063D+09

441

442 -----

443 Gamma Lines at end of Simulation 0.2600000000E+06

444 -----

Z	A	M	Gamma Energy/KeV	Gamma count/s
445	24	51	0	0.3200824000E+03
446				0.1432815997E+04
447	25	52	0	0.2005800000E+03
448				0.3428182982E+02
449	25	52	0	0.3460199000E+03
				0.4420551289E+03
				0.3980899000E+03
				0.4014582702E+02

450	25	52	0	0.3995700000E+03	0.8254703759E+02
451	25	52	0	0.5020600000E+03	0.9472610871E+02
452	25	52	0	0.6001600000E+03	0.1759199162E+03
453	25	52	0	0.6474699000E+03	0.1804306833E+03
454	25	52	0	0.7442329000E+03	0.4059690373E+05
455	25	52	0	0.8481800000E+03	0.1497574671E+04
456	25	52	0	0.9018899000E+03	0.1984737065E+02
457	25	52	0	0.9355440000E+03	0.4262674892E+05
458	25	52	0	0.1045750000E+04	0.3157536957E+02
459	25	52	0	0.1246278000E+04	0.1899032941E+04
460	25	52	0	0.1247879000E+04	0.1714091491E+03
461	25	52	0	0.1333648000E+04	0.2286958910E+04
462	25	52	0	0.1434092000E+04	0.4510767081E+05
463	25	52	0	0.1645819000E+04	0.2120060528E+02
464	25	52	0	0.1981120000E+04	0.1533660808E+02
465	25	52	0	0.2257419000E+04	0.1217907112E+01
466	25	54	0	0.8348479000E+03	0.9736656444E+04
467	26	52	0	0.1686880000E+03	0.1874526871E+03
468	26	52	0	0.3777479000E+03	0.3106047431E+01
469	26	52	0	0.1039928000E+04	0.1798237987E+00
470	26	55	0	0.1259999000E+03	0.1639505194E-04
471	27	55	0	0.9190000000E+02	0.6907616941E+03
472	27	55	0	0.3853999000E+03	0.3208699482E+03
473	27	55	0	0.4114999000E+03	0.6372833694E+03
474	27	55	0	0.4771999000E+03	0.1198805779E+05
475	27	55	0	0.5199999000E+03	0.4902179764E+03
476	27	55	0	0.8036999000E+03	0.1109675238E+04
477	27	55	0	0.8270000000E+03	0.1247827576E+03
478	27	55	0	0.9311000000E+03	0.4456527059E+05
479	27	55	0	0.9845999000E+03	0.3075003670E+03
480	27	55	0	0.1212799000E+04	0.1559783876E+03
481	27	55	0	0.1316600000E+04	0.4211417476E+04
482	27	55	0	0.1369999000E+04	0.1733589026E+04
483	27	55	0	0.1408499000E+04	0.1002718588E+05
484	27	55	0	0.1555999000E+04	0.2718481506E+02
485	27	55	0	0.1622299000E+04	0.2673916235E+02
486	27	55	0	0.1792100000E+04	0.4857614494E+02
487	27	55	0	0.1940599000E+04	0.8467401411E+01
488	27	55	0	0.2144200000E+04	0.5347832470E+02
489	27	55	0	0.2177600000E+04	0.1738045553E+03
490	27	55	0	0.2578700000E+04	0.2540220423E+02
491	27	55	0	0.2872400000E+04	0.6996747482E+02
492	27	55	0	0.2938900000E+04	0.3386960565E+02
493	27	55	0	0.3108300000E+04	0.3119568941E+01
494	27	56	0	0.2634099000E+03	0.2075380569E+01
495	27	56	0	0.4113799000E+03	0.2358388082E+01
496	27	56	0	0.4865399000E+03	0.5754466920E+01
497	27	56	0	0.6550000000E+03	0.3584749885E+01
498	27	56	0	0.6746999000E+03	0.3584749885E+01
499	27	56	0	0.7335109000E+03	0.1792374942E+02
500	27	56	0	0.7877419000E+03	0.2971568983E+02
501	27	56	0	0.8467709000E+03	0.9427420519E+04
502	27	56	0	0.8527800000E+03	0.4716776164E+01
503	27	56	0	0.8965309000E+03	0.8112855002E+01
504	27	56	0	0.9773729000E+03	0.1366921732E+03
505	27	56	0	0.9973299000E+03	0.1216928250E+02
506	27	56	0	0.1037839000E+04	0.1336733422E+04
507	27	56	0	0.1089030000E+04	0.4716776164E+01
508	27	56	0	0.1140404000E+04	0.1292396669E+02
509	27	56	0	0.1159919000E+04	0.8961874712E+01
510	27	56	0	0.1175101000E+04	0.2158396773E+03

511	27	56	0	0.1198780000E+04	0.4811111687E+01
512	27	56	0	0.1238281000E+04	0.6311046508E+04
513	27	56	0	0.1272200000E+04	0.2264052559E+01
514	27	56	0	0.1335389000E+04	0.1113159175E+02
515	27	56	0	0.1360215000E+04	0.4046993949E+03
516	27	56	0	0.1442750000E+04	0.1745207181E+02
517	27	56	0	0.1462340000E+04	0.6131809013E+01
518	27	56	0	0.1640404000E+04	0.6792156733E+01
519	27	56	0	0.1771351000E+04	0.1459370545E+04
520	27	56	0	0.1810771000E+04	0.6018606385E+02
521	27	56	0	0.1963714000E+04	0.6829891886E+02
522	27	56	0	0.2015181000E+04	0.2867799908E+03
523	27	56	0	0.2034755000E+04	0.7443072787E+03
524	27	56	0	0.2113123000E+04	0.3547015675E+02
525	27	56	0	0.2212933000E+04	0.3726253170E+02
526	27	56	0	0.2276360000E+04	0.1207494698E+02
527	27	56	0	0.2373700000E+04	0.7735512909E+01
528	27	56	0	0.2523860000E+04	0.6414815583E+01
529	27	56	0	0.2598459000E+04	0.1632004553E+04
530	27	56	0	0.2657400000E+04	0.1981045989E+01
531	27	56	0	0.3009595000E+04	0.1094292070E+03
532	27	56	0	0.3201962000E+04	0.3131939373E+03
533	27	56	0	0.3253415000E+04	0.7660044491E+03
534	27	56	0	0.3272990000E+04	0.1820674656E+03
535	27	56	0	0.3369690000E+04	0.1037689813E+01
536	27	56	0	0.3451152000E+04	0.9169411920E+02
537	27	56	0	0.3547930000E+04	0.1886710466E+02
538	27	56	0	0.3600700000E+04	0.1556536134E+01
539	27	56	0	0.3611799000E+04	0.8018519479E+00
540	27	57	0	0.1441290000E+02	0.9899856117E+01
541	27	57	0	0.1220606000E+03	0.9251393926E+02
542	27	57	0	0.1364735000E+03	0.1154261619E+02
543	27	57	0	0.3396899000E+03	0.3998850178E-02
544	27	57	0	0.3523300000E+03	0.3242310955E-02
545	27	57	0	0.3667999000E+03	0.1296924382E-02
546	27	57	0	0.5700900000E+03	0.1707617103E-01
547	27	57	0	0.6924099000E+03	0.1610347774E+00
548	27	57	0	0.7065399000E+03	0.5403851592E-02
549	27	58	0	0.8107592000E+03	0.2431078541E+02
550	27	58	0	0.8639510000E+03	0.1686720915E+00
551	27	58	0	0.1674725000E+04	0.1271152178E+00

552  
553 -----

554 Activities Ordered by Most Active

555 -----  
556

557	15985	CO	27	55	0	0.5942036078E+05
558	14802	MN	25	52	0	0.4510767081E+05
559	14211	CR	24	51	0	0.1444370965E+05
560	15395	FE	26	55	0	0.1280863433E+05
561	14804	MN	25	54	0	0.9738993802E+04
562	15986	CO	27	56	0	0.9433552328E+04
563	15392	FE	26	52	0	0.1890116552E+03
564	13619	V	23	49	0	0.1313773643E+03
565	15987	CO	27	57	0	0.1080770318E+03
566	15988	CO	27	58	0	0.2444523420E+02
567	593	H	1	3	0	0.1615640588E+02
568	14803	MN	25	53	0	0.1176009917E-02

569  
570 -----

571 Totals 1.8880

572 -----  
  
573  
574 Total Activity/Bq: 0.1514219906E+06  
575 Total Gamma Power/eV/s: 0.2559334931E+12  
576 Total Gamma Power/Watts: 0.4100506435E-07  
577 Absorbed Dose\*/Grays/s: 0.4078849177E-10  
578 Absorbed Dose\*/Grays/hr: 0.1468385704E-06  
579 Fraction of annual dosage if exposed for 1 hr: 0.1468385634E-03  
580  
581 Absorbed dose, assumes all energy absorbed, 80kg human, 1m from point-target, 1m surface area exposed to  
irradiation.  
582  
583 Dose Limits:  
584 employees 18+ 20 millisieverts/year  
585 trainees 18+ 6 millisieverts/year  
586 public and under 18s 1 millisieverts/year  
587 public and under 18s millisieverts averaged per hour: 0.1140771128E-03  
588 Dose averaged over area of skin not exceeding 1cm<sup>2</sup>  
589 Source: <http://www.hse.gov.uk/radiation/ionising/doses/>  
590  
591  
592  
593 -----  
  
594 Activity calculation complete. 1.8880  
595 -----

---

## **Appendix E**

### **Activity V2 Full Results**

## E.1 36MeV Proton Irradiated Iron

### E.1.1 Calculation Settings

Material	Pure Iron
Sheet thickness	0.5mm
Density	7,808 $kgm^3$
Beam energy	36MeV
Beam projectile	proton
Beam current	0.5 microamps
Beam duration	300s
Simulation end time	260,000s

Table E.1

### E.1.2 Results File

Listing E.1: Isotope Activity at End of Simulation

```

1 13.130 #####
2 13.130 Sim 1
3 13.130 #####
4 13.130 #####
5 13.130 Beam flux: 3120754564499.886
6 13.130 Target depth: 0.0005
7 13.130 Projectile: p
8 13.130 Beam duration/s: 300.0
9 13.130 Sim end time/s: 260000.0
10 13.130
11 13.130 Starting Target Material:
12 13.130
13 13.130 #####
14 13.130 Isotopes
15 13.130 #####
16 13.130 Isotope Percentage by Mass
17 13.130 26 Fe 56 91.753250
18 13.130 26 Fe 54 5.845279
19 13.130 26 Fe 57 2.119074
20 13.130 26 Fe 58 0.282397
21 13.130
22 13.130 Loading exyz /cloud/Code/python/activity/examples/fe/Fe36MeV.exyz
23 13.620 Load complete
24 13.620
25 13.620 Calculate residual reaction rates
26 25.746 Save reaction rates
27 25.746
28 25.746
29 25.746 #####
30 25.746 Particle & Residual Reaction Rates
31 25.746 #####
32 25.746 Source isotope: 54Fe (26054)
33 25.746 helium-3: 8128226.96053868
34 25.746 deuteron: 31267425.752861295
35 25.746 triton: 658063.9037544213
36 25.746 alpha: 47929259.98417321
37 25.746 gamma: 529302786.1997857
38 25.746 proton: 1280534004.9887292

```

39 25.746 neutron: 341994739.706016  
 40 25.746 52Fe: 10375758.270700272  
 41 25.746 54Co(meta 1): 1268489.4293415854  
 42 25.746 52Fe(meta 1): 91996.5249677117  
 43 25.746 53Fe: 109537376.08734141  
 44 25.746 53Fe(meta 1): 7878234.293365053  
 45 25.746 50Mn(meta 1): 447309.355049635  
 46 25.746 54Co: 6134948.952811248  
 47 25.746 50Mn: 405358.7041290127  
 48 25.746 52Mn(meta 1): 51514257.29426319  
 49 25.746 52Mn: 79833423.71247151  
 50 25.746 54Fe: 92416418.45220399  
 51 25.746 55Co: 140959.1816623515  
 52 25.746 47V: 128374.10459496645  
 53 25.746 53Mn: 121917363.90080762  
 54 25.746 49Cr: 2248642.2236076766  
 55 25.746 53Co: 166474.72453697983  
 56 25.746 51Mn: 13144593.549430655  
 57 25.746 51Cr: 2004868.7752378457  
 58 25.746 52Cr: 39343113.4701649  
 59 25.746 46Ti: 10113.753559680361  
 60 25.746 50Cr: 20277932.722341496  
 61 25.746 49V: 4456332.156319902  
 62 25.746  
 63 25.746  
 64 25.746 Source isotope: 56Fe (26056)  
 65 25.746 helium-3: 113796332.11233293  
 66 25.746 deuteron: 640048071.8232051  
 67 25.746 triton: 27462082.512083083  
 68 25.746 alpha: 784412097.0717504  
 69 25.746 gamma: 12750097788.074738  
 70 25.746 proton: 12862634835.97489  
 71 25.746 neutron: 13361355665.752747  
 72 25.746 53Fe(meta 2): 35110.96922030936  
 73 25.746 54Co(meta 1): 2098133.50324677  
 74 25.746 48Ti: 130153.19806252541  
 75 25.746 53Fe: 1650122.878507311  
 76 25.746 48Ti(meta 2): 1.3498721704564602  
 77 25.747 54Co: 2217228.5579557572  
 78 25.747 52Mn(meta 1): 51722330.55364108  
 79 25.747 52Mn: 69812825.68860963  
 80 25.747 53Mn: 617186326.652764  
 81 25.747 56Co: 297046346.5343496  
 82 25.747 57Co: 3332169.8943027295  
 83 25.747 56Fe: 1324989311.7401316  
 84 25.747 54Mn: 1105697455.8025718  
 85 25.747 48V: 6824.981311026873  
 86 25.747 54Cr: 8886829.249919416  
 87 25.747 51V: 1908704.2981555585  
 88 25.747 54Fe: 1768658520.9905515  
 89 25.747 55Mn: 469054279.81113464  
 90 25.747 51Mn: 18427975.828979623  
 91 25.747 51Cr: 157917501.71033847  
 92 25.747 55Co: 158802991.55014923  
 93 25.747 55Fe: 3310150310.2897525  
 94 25.747 52Cr: 198718293.6955533  
 95 25.747 53Cr: 868047.1046782847  
 96 25.747 50Cr: 24883.064209825417  
 97 25.747 49V: 3730773.4294895227  
 98 25.747  
 99 25.747

100	25.747	Source isotope: 57Fe (26057)
101	25.747	helium-3: 2815186.114818401
102	25.747	deuteron: 16795071.729165718
103	25.747	triton: 1290070.0147642847
104	25.747	alpha: 18996008.688927133
105	25.747	gamma: 318048823.03063726
106	25.747	proton: 263457267.1862257
107	25.747	neutron: 381235458.34554696
108	25.747	58Co(meta 1): 17711.056288183212
109	25.747	58Co: 45119.123356589305
110	25.747	52Mn(meta 1): 756983.0649500752
111	25.747	52Mn: 1130057.9800183107
112	25.747	54Mn: 10749922.705687553
113	25.747	57Co: 6645538.449427711
114	25.747	57Fe: 22282747.27841167
115	25.747	53Mn: 9077540.382290483
116	25.747	56Mn: 5561497.664552677
117	25.747	54Cr: 32475.710066264975
118	25.747	51V: 278.34482695928915
119	25.747	50V: 60535.34442859975
120	25.747	56Co: 7513776.276815263
121	25.747	54Fe: 2335917.272526654
122	25.747	49Ti: 66.6465837718329
123	25.747	55Mn: 17896651.547242593
124	25.747	55Cr: 5259.683969248049
125	25.747	51Cr: 3777.5330594318025
126	25.747	55Co: 1040282.7451955614
127	25.747	55Fe: 52180477.480709895
128	25.747	52Cr: 3736281.5484410194
129	25.747	53Cr: 810918.822403883
130	25.747	52V: 1714.7498473523403
131	25.747	56Fe: 82701492.44833918
132	25.747	49V: 29701.937810047228
133	25.747	
134	25.747	
135	25.747	Source isotope: 58Fe (26058)
136	25.747	helium-3: 265562.0411442721
137	25.747	deuteron: 2046736.073932219
138	25.747	triton: 161109.31102851706
139	25.747	alpha: 2132323.5017096302
140	25.747	gamma: 43596863.655506425
141	25.747	proton: 27781338.358164437
142	25.747	neutron: 60156638.64809996
143	25.747	58Co(meta 1): 393025.0417438573
144	25.747	58Co: 787138.5782124916
145	25.747	58Fe: 3578387.673606792
146	25.747	59Co: 7298.027151484427
147	25.747	55Mn: 917307.4059246042
148	25.747	53Mn: 519923.8215990508
149	25.747	50Ti: 139.09122070067937
150	25.747	56Mn: 495624.1780436759
151	25.747	54Mn: 750596.5413600848
152	25.747	54Cr: 42781.814693838154
153	25.747	51V: 3640.3531430735275
154	25.747	50V: 610.1198082776706
155	25.747	56Co: 528863.6120293873
156	25.747	54Fe: 188.91955540759065
157	25.747	57Co: 2188175.452108935
158	25.747	57Fe: 7076618.451591522
159	25.747	55Cr: 26.068658806358545
160	25.747	57Mn: 347080.97455489024

```

161 25.747 55Co:          3843.437575999017
162 25.747 55Fe:          445703.4711783764
163 25.747 52Cr:          1584.071467022736
164 25.747 53Cr:          94533.22614130769
165 25.747 56Fe:          8672082.613410642
166 25.747
167 25.747
168 25.747 Calculate residual amounts over time
169 25.747
170 25.747 In beam time steps: 101
171 25.747 Out of beam time steps: 201
172 25.747
173 25.748 In beam activity over 101 time steps
174 27.168 In beam activity over 201 time steps
175 30.417 Tally amount by residual isotope
176 30.417 Tally amount of residual isotopes by target isotope
177 30.417 Activity per target isotope by residual radioactive isotope over time
178 30.419 Activity by residual radioactive isotope over time
179 30.420 Activity resulting from each target isotope over time
180 30.422 Total activity from all targets and residuals over time
181 30.424 Activity per target and residual over time
182 30.528 Activity per residual over time
183 30.655 Activity per target over time
184 30.771 Activity data for pie charts
185 30.793 Tally gammas by target -> residual -> energy
186 30.797 Tally gammas by residual -> energy
187 30.800 Tally gammas by target -> energy
188 30.803 Tally gammas -> by gamma energy
189 30.807 Tally gammas by residual and gamma energies
190 30.809 Tally gammas by target and gamma energies
191 30.811 Tally gammas by target, residual and gamma energies
192 30.814 Per target and residual gamma lines over time
193 31.061 Per residual gamma lines over time
194 31.297 Per target gamma lines over time
195 31.542 Tallied gamma lines over time
196 31.790 Per residual gamma lines over time
197 31.966 Per target gamma lines over time
198 32.119 Total activity and energy of gammas over time
199 32.266 Gamma data for pie charts
200 32.294 Activity and Gamma totals over time for charts
201 32.294
202 32.294
203 45.551 Gamma Dose - Beam End
204 45.551 =====
205 45.551 Activity (Bq)      1.1873e+08
206 45.551 Gamma Energy (eV)   9.7722e+13
207 45.551 Gamma Energy (mW)   1.5657e-02
208 45.551 Absorbed Dose (mGy/s) 1.5570e-11
209 45.551 Absorbed Dose (mGy/hr) 5.6051e-08
210 45.551
211 45.552
212 45.552 Gamma Dose - Sim End
213 45.552 =====
214 45.552 Activity (Bq)      1.8420e+05
215 45.552 Gamma Energy (eV)   3.5639e+11
216 45.552 Gamma Energy (mW)   5.7100e-05
217 45.552 Absorbed Dose (mGy/s) 5.6782e-14
218 45.552 Absorbed Dose (mGy/hr) 2.0442e-10
219 45.552
220 45.552
221 45.552 Absorbed Dose Calculations

```

222 45.552 =====  
223 45.552 Absorbed dose assumptions:  
224 45.552 1. radiation from point, emitted isotropically  
225 45.552 2. 80Kg human  
226 45.552 3. 1m from point source  
227 45.552 4. 1m squared surface area  
228 45.552 5. all energy absorbed  
229 45.552  
230 45.552  
231 45.552 Dose Limits  
232 45.552 =====  
233 45.552 employees 18+ 20 millisieverts/year  
234 45.552 trainees 18+ 6 millisieverts/year  
235 45.552 public and under 18s 1 millisievert/year  
236 45.552 public and under 18s 1.140771128E-04 millisieverts/hour  
237 45.552 public and under 18s 3.2E-08 millisieverts/second  
238 45.552  
239 45.552 Dose averaged over area of skin not exceeding 1cm<sup>2</sup>  
240 45.552 Source: <http://www.hse.gov.uk/radiation/ionising/doses/>  
241 45.552  
242 45.552 1mW gammas completely absorbed by 80kg human will give public limit for entire year over 80 seconds.  
243 45.552  
244 45.552 Dose under 1W gammas absorbed for 1s no immediate changes (under 100 Rads)  
245 45.552 Dose 1W-2W gammas absorbed for 1s acute radiation syndrome (100-200 Rads)  
246 45.552 Dose 10W+ gammas absorbed for 1s mostly fatal (1000+ Rads)

---

### E.1.3 Activity Tallied by Residual Isotope

Residual activity at time t=3.0  
Total activity: 13658991.008165512

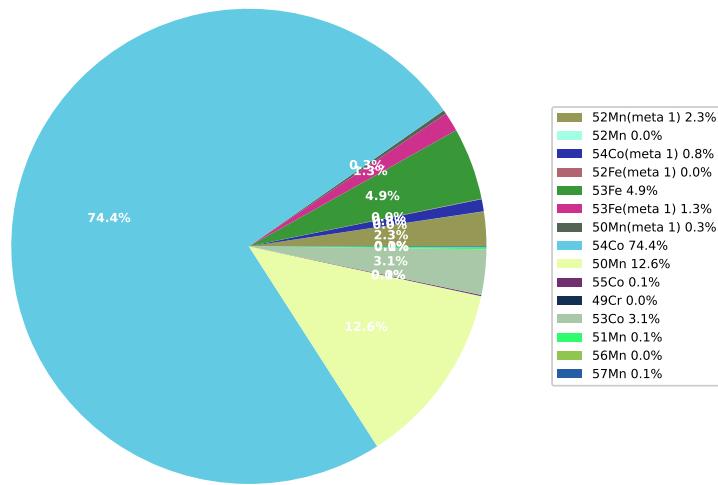


Figure E.1

Residual activity at time t=60.0  
Total activity: 38065946.53520206

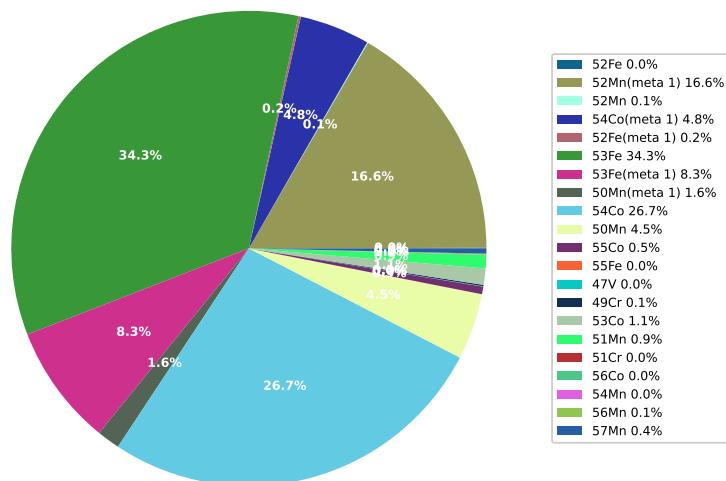


Figure E.2

Residual activity at time t=300.0  
Total activity: 118728909.78631218

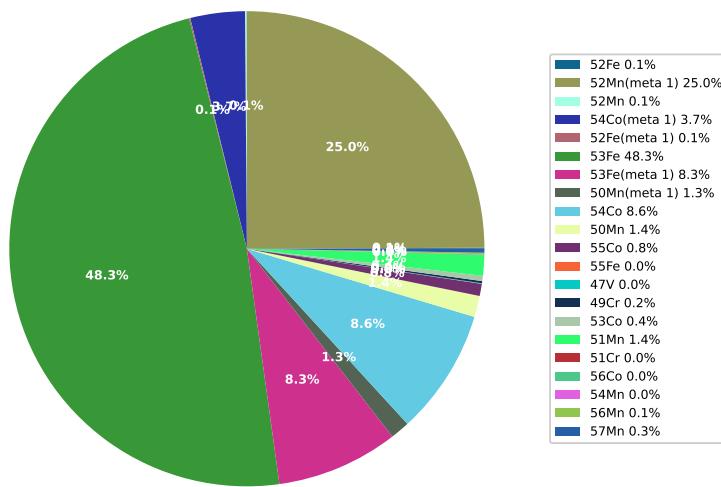


Figure E.3

Residual activity at time t=86001.0  
Total activity: 554898.1962193253

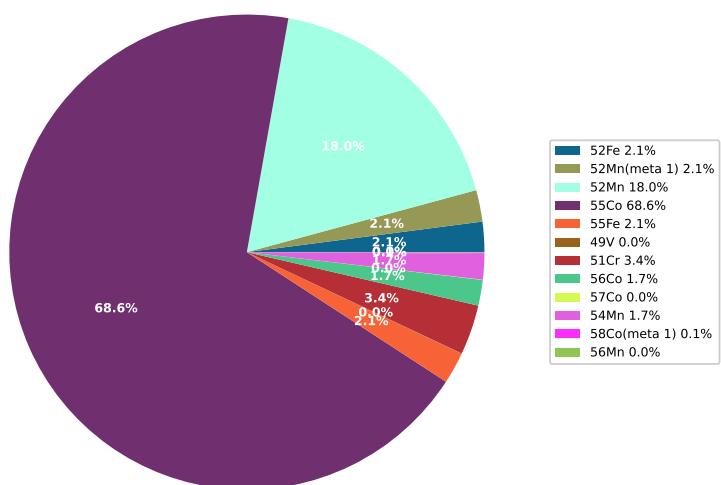


Figure E.4

Residual activity at time t=173000.5  
Total activity: 287692.0694298974

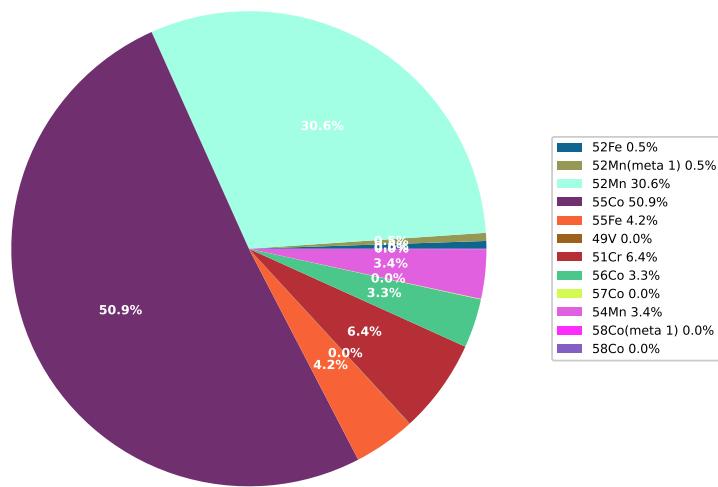


Figure E.5

Residual activity at time t=260000.0  
Total activity: 184007.52249499888

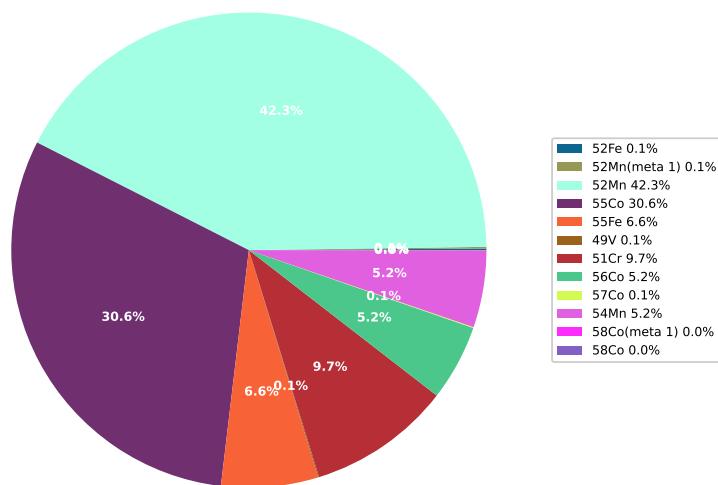


Figure E.6

### E.1.4 Gamma Output Tallied by Residual Isotope

Residual gamma power at time t=3.0  
Total gamma power: 0.00024354619226584488 mW

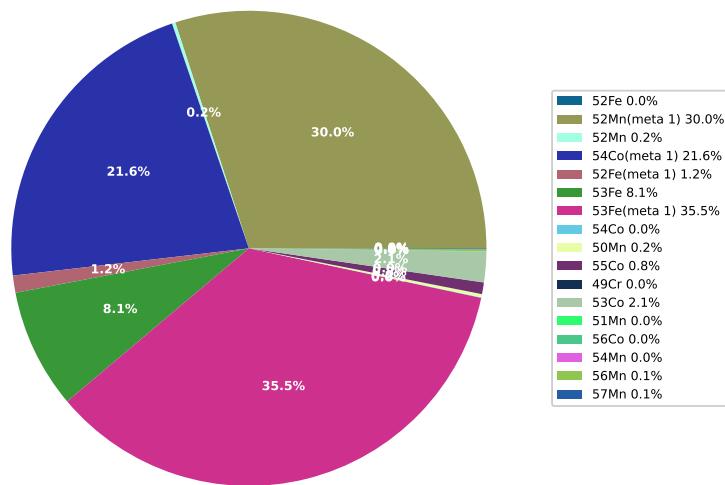


Figure E.7

Residual gamma power at time t=60.0  
Total gamma power: 0.004306486573204578 mW

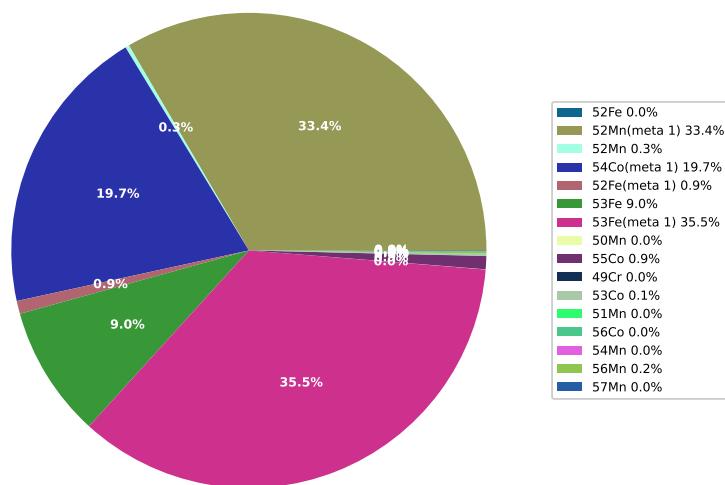


Figure E.8

Residual gamma power at time t=300.0  
Total gamma power: 0.015656781568754878 mW

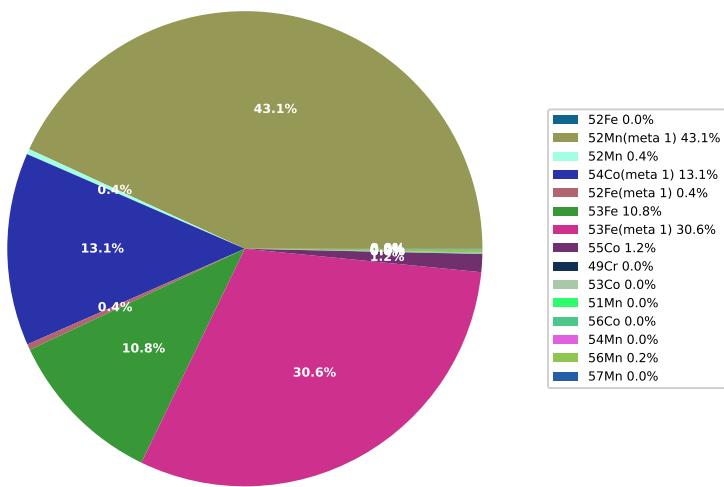


Figure E.9

Residual gamma power at time t=86001.0  
Total gamma power: 0.00013470195336474222 mW

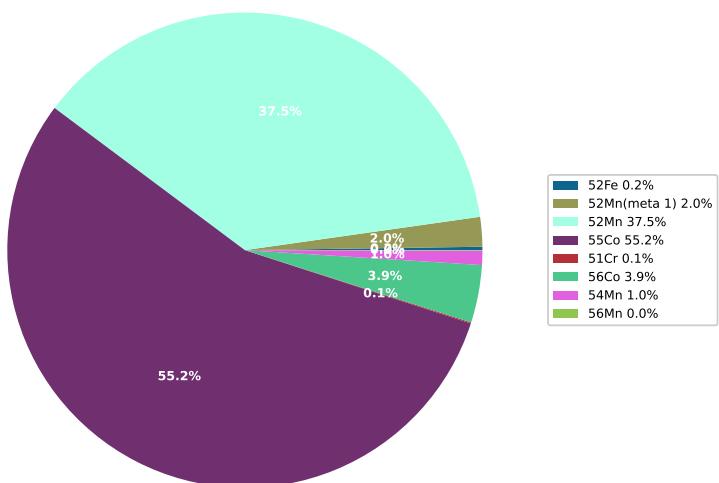


Figure E.10

Residual gamma power at time t=173000.5  
 Total gamma power: 8.027298909252732e-05 mW

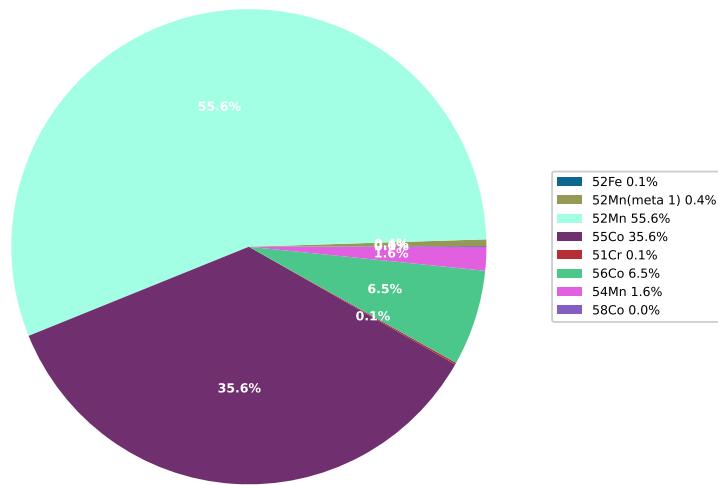


Figure E.11

Residual gamma power at time t=260000.0  
 Total gamma power: 5.703563656739926e-05 mW

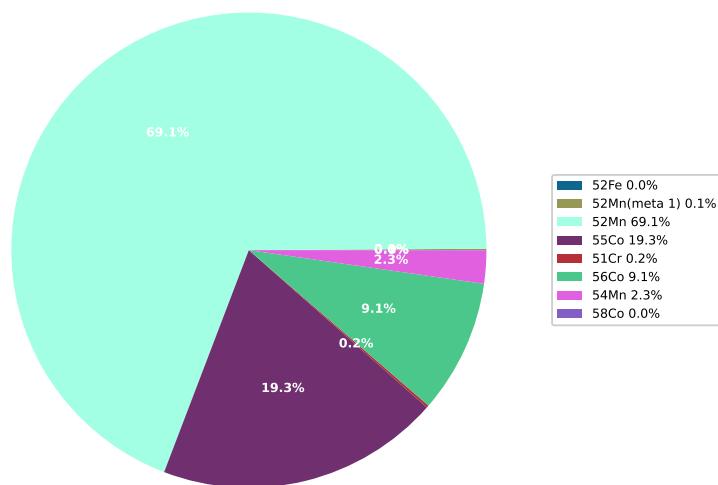


Figure E.12

### E.1.5 Activity Tallied by Target Isotope

Target activity at time t=3.0  
Total activity: 13658991.008165512

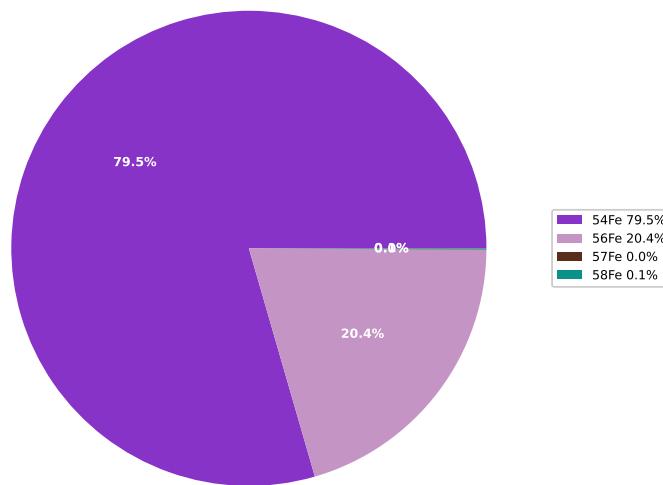


Figure E.13

Target activity at time t=60.0  
Total activity: 38065946.53520206

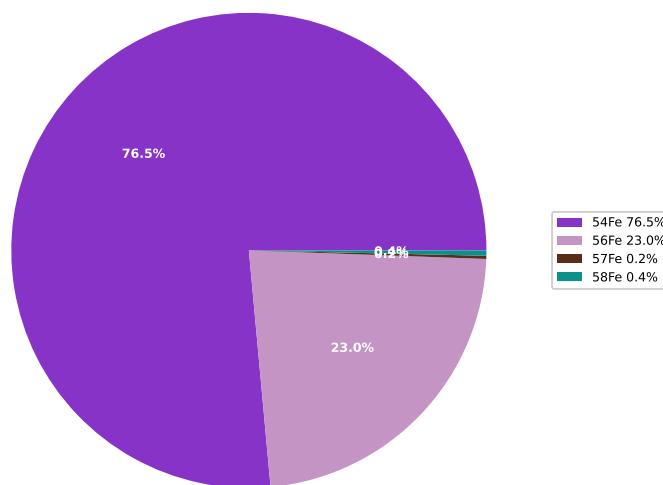


Figure E.14

Target activity at time t=300.0  
Total activity: 118728909.78631218

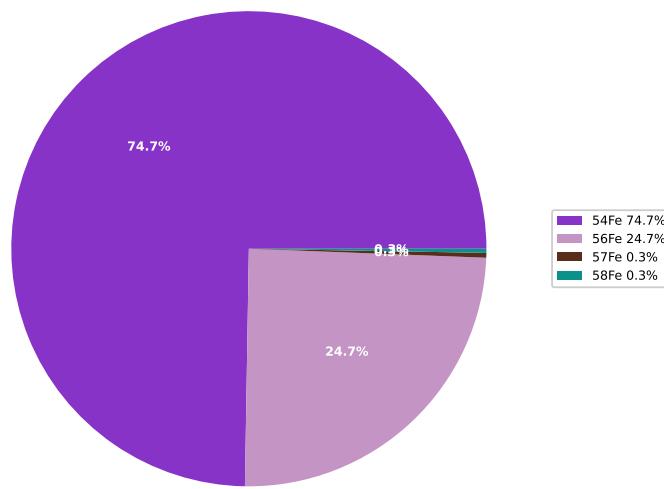


Figure E.15

Target activity at time t=86001.0  
Total activity: 554898.1962193253

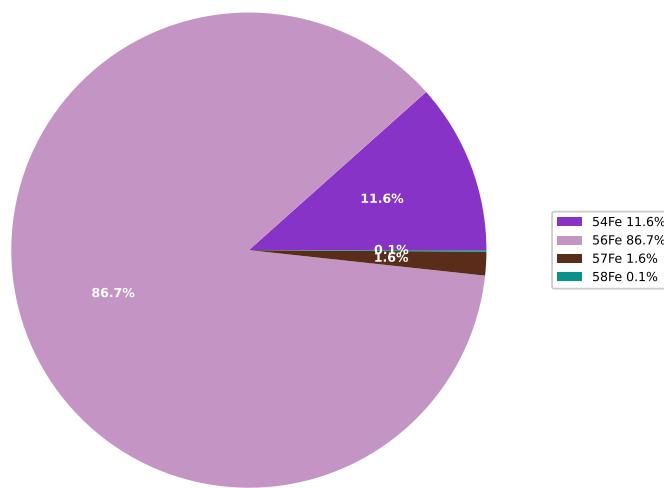


Figure E.16

Target activity at time  $t=173000.5$   
Total activity: 287692.0694298974

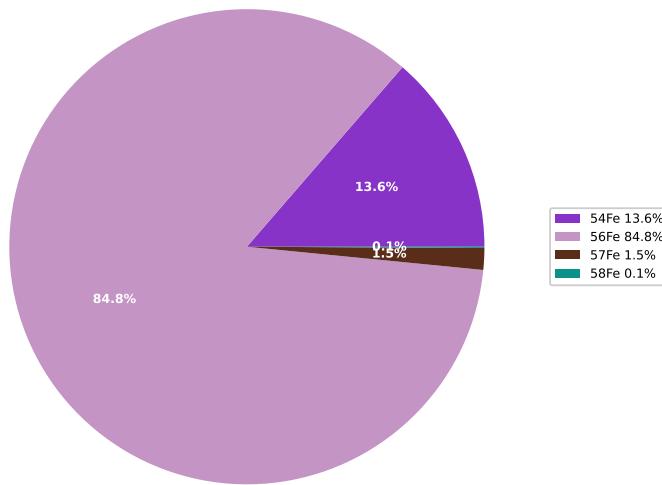


Figure E.17

Target activity at time  $t=260000.0$   
Total activity: 184007.52249499888

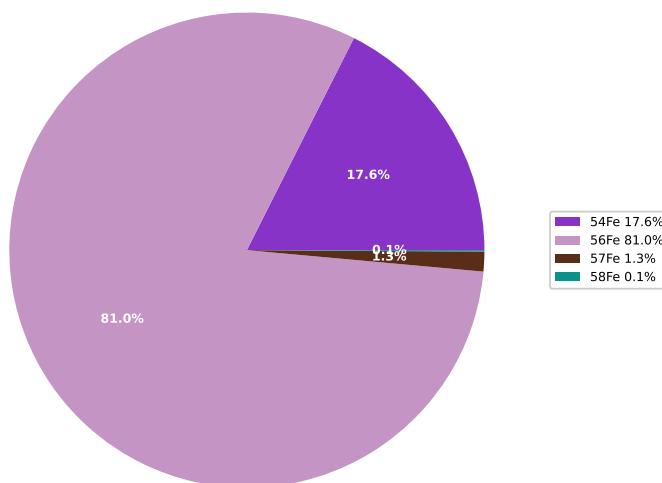


Figure E.18

### E.1.6 Gamma Output Tallied by Target Isotope

Target gamma power at time t=3.0  
Total gamma power: 0.00024354619226584488 mW

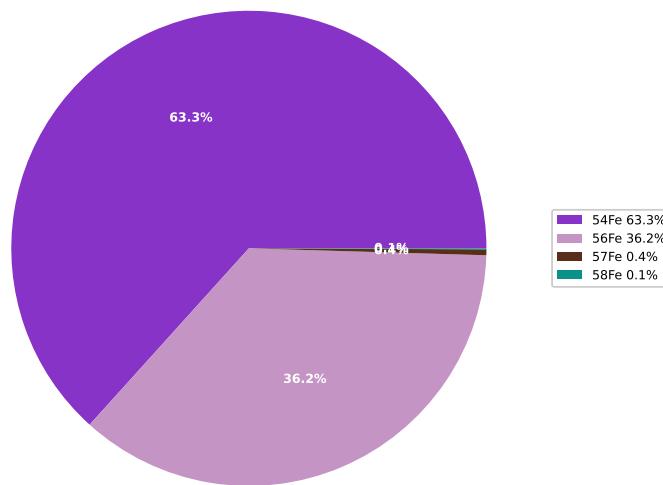


Figure E.19

Target gamma power at time t=60.0  
Total gamma power: 0.004306486573204578 mW

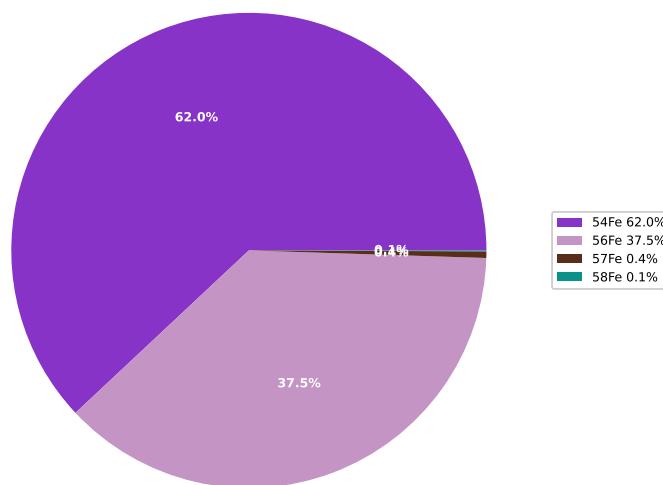


Figure E.20

Target gamma power at time t=300.0  
Total gamma power: 0.015656781568754878 mW

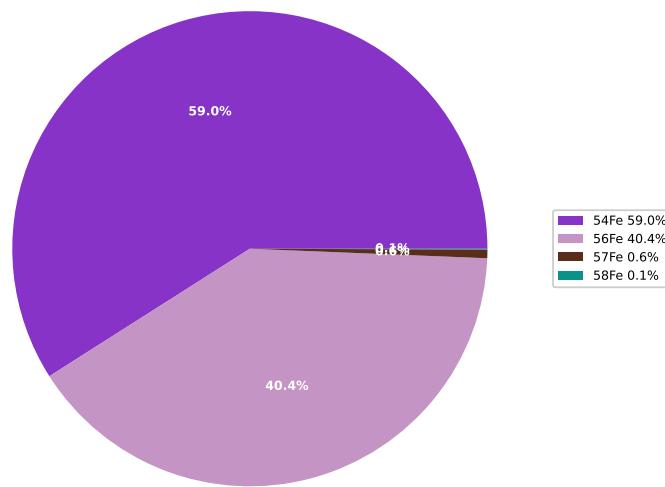


Figure E.21

Target gamma power at time t=86001.0  
Total gamma power: 0.00013470195336474222 mW

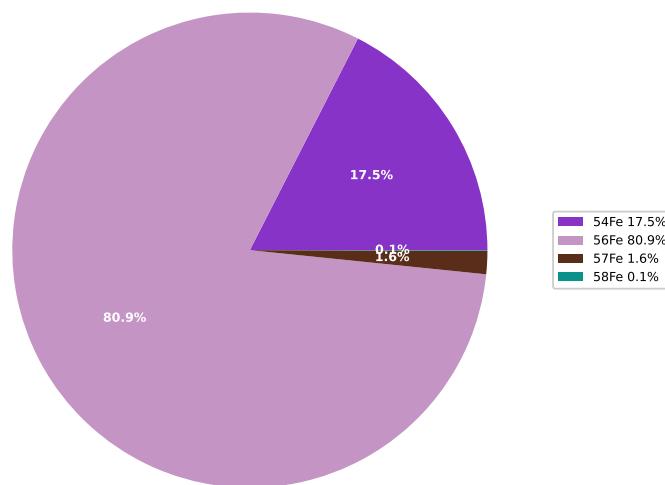


Figure E.22

Target gamma power at time t=173000.5  
Total gamma power: 8.027298909252732e-05 mW

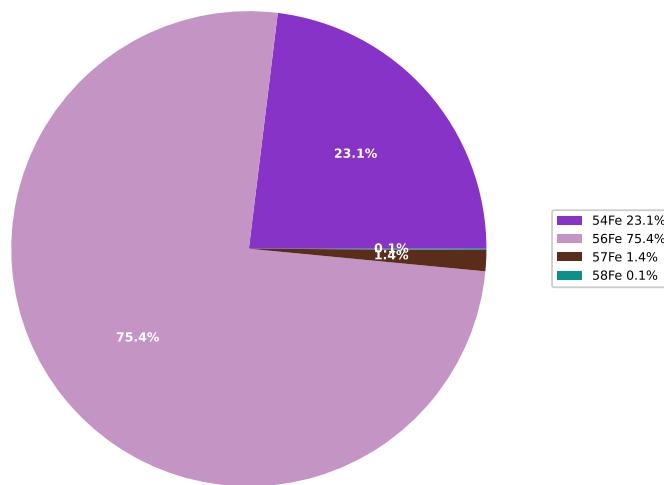


Figure E.23

Target gamma power at time t=260000.0  
Total gamma power: 5.703563656739926e-05 mW

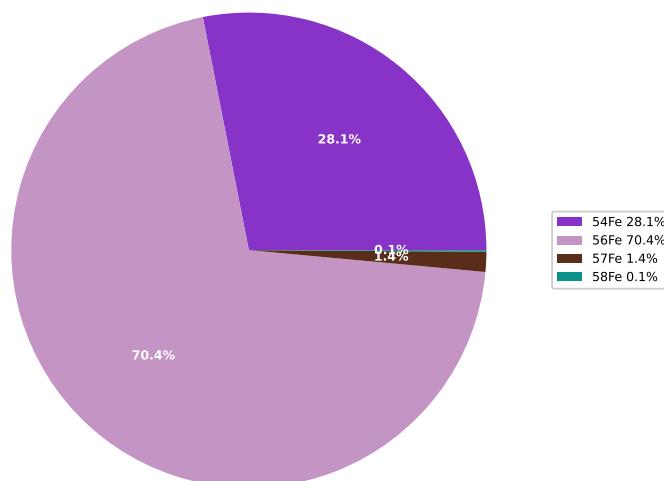


Figure E.24

## E.2 13MeV Proton Irradiated Molybdenum

### E.2.1 Calculation Settings

Material	Pure Molybdenum
Sheet thickness	506 micrometers
Density	10,220 $kgm^3$
Beam energy	13MeV
Beam projectile	proton
Beam current	5.0 microamps
Beam duration	1500s
Simulation end time	260,000s

Table E.2

### E.2.2 Results

Listing E.2: Final results for Molybdenum Irradiation

```

1 23.132 #####
2 23.132 Sim 1
3 23.132 #####
4 23.132 #####
5 23.132 Beam flux:      31207545644998.863
6 23.132 Target depth:    0.000506
7 23.132 Projectile:     p
8 23.132 Beam duration/s: 1500.0
9 23.132 Sim end time/s: 606300.0
10 23.132 #####
11 23.132 Starting Target Material:
12 23.132 #####
13 23.132 #####
14 23.132           Isotopes
15 23.132 #####
16 23.132   Isotope      Percentage by Mass
17 23.132   42 Mo 98     24.127937
18 23.132   42 Mo 96     16.678431
19 23.132   42 Mo 95     15.917894
20 23.132   42 Mo 92     14.840324
21 23.132   42 Mo 100    9.630239
22 23.132   42 Mo 97     9.555955
23 23.132   42 Mo 94     9.249221
24 23.132 #####
25 196.630 #####
26 196.630 #####
27 196.630 Particle & Residual Reaction Rates
28 196.630 #####
29 196.630 Source isotope: 92Mo (42092)
30 196.630 #####
31 196.630 #####
32 196.630 Source isotope: 94Mo (42094)
33 196.630 helium-3:      3.1332823136574574e-12
34 196.630 deuteron:      26779.397771044376
35 196.630 triton:        4.691618589935858
36 196.630 alpha:         35497160.708963595
37 196.630 gamma:         4046256131.9846444
38 196.630 proton:        474276321.7635846

```

39 196.630 neutron: 2939219987.127653  
 40 196.630 94Tc(meta 1): 982541746.5227002  
 41 196.630 95Tc(meta 1): 3220288.2361354767  
 42 196.630 95Tc: 4316294.073460804  
 43 196.630 91Nb: 16990624.30575451  
 44 196.630 91Nb(meta 1): 18484514.956021547  
 45 196.630 94Tc: 1914038155.7223403  
 46 196.630 93Mo: 42473999.299622655  
 47 196.630 93Mo(meta 1): 126.22443599491  
 48 196.630 94Mo: 431701273.0135327  
 49 196.630 90Zr: 18272.130848077595  
 50 196.630  
 51 196.630  
 52 196.630 Source isotope: 95Mo (42095)  
 53 196.630 helium-3: 5.46887072783639e-07  
 54 196.630 deutron: 1234961.9156552393  
 55 196.630 triton: 41.10937959624232  
 56 196.630 alpha: 54324719.08951578  
 57 196.630 gamma: 9102804904.23539  
 58 196.630 proton: 894337849.3520026  
 59 196.630 neutron: 5518562890.307192  
 60 196.630 95Tc(meta 1): 1218252833.6464837  
 61 196.630 92Nb(meta 1): 33006155.92287399  
 62 196.630 96Tc(meta 1): 2679422.9964109  
 63 196.630 95Tc: 3689625934.027424  
 64 196.630 92Nb: 14122020.720040252  
 65 196.631 96Tc: 4182516.45184412  
 66 196.631 94Tc(meta 1): 13905093.155669797  
 67 196.631 91Nb: 6611098.919676594  
 68 196.631 91Nb(meta 1): 576457.3002812921  
 69 196.631 94Tc: 59984103.84506015  
 70 196.631 95Mo: 439173380.2083463  
 71 196.631 94Mo: 456055572.20441234  
 72 196.631  
 73 196.631  
 74 196.631 Source isotope: 96Mo (42096)  
 75 196.631 helium-3: 5.5873148279970936e-15  
 76 196.631 deutron: 89980.87611341325  
 77 196.631 triton: 146.48193254666154  
 78 196.631 alpha: 16359850.77593771  
 79 196.631 gamma: 11880347277.430742  
 80 196.631 proton: 319281355.5874101  
 81 196.631 neutron: 6458872187.0255995  
 82 196.631 93Nb: 10952560.277578255  
 83 196.631 97Tc: 5027859.164271742  
 84 196.631 96Tc(meta 1): 2314902593.6163425  
 85 196.631 97Tc(meta 1): 2657069.708104688  
 86 196.631 96Tc: 2988618887.151654  
 87 196.631 93Nb(meta 1): 5392146.865275863  
 88 196.631 89Y: 4.699677647993188  
 89 196.631 89Y(meta 1): 0.024318699111099025  
 90 196.631 95Tc(meta 1): 186850567.4939038  
 91 196.631 92Nb(meta 1): 2991.5174685434013  
 92 196.631 95Tc: 387490658.45324874  
 93 196.631 92Nb: 2926.3241012165836  
 94 196.631 96Mo: 313532526.5135967  
 95 196.631 95Mo: 5619671.213466896  
 96 196.631  
 97 196.631  
 98 196.631 Source isotope: 97Mo (42097)  
 99 196.631 helium-3: 2.6147848182753646e-08

100 196.631 deuteron: 1162205.9105530165  
 101 196.631 triton: 185.73563697784863  
 102 196.631 alpha: 10097428.581276963  
 103 196.631 gamma: 6269889067.613607  
 104 196.631 proton: 197692275.39840198  
 105 196.631 neutron: 4202157441.703513  
 106 196.631 97Tc: 2012907421.2696567  
 107 196.631 94Nb(meta 1): 7124606.3063902855  
 108 196.631 97Tc(meta 1): 515939120.74455863  
 109 196.631 90Y: 0.0007502229001053001  
 110 196.631 94Nb: 2737836.422765725  
 111 196.631 90Y(meta 1): 5.501140264263349e-05  
 112 196.631 93Nb: 173177.71086113452  
 113 196.631 96Tc(meta 1): 293382016.0592189  
 114 196.631 96Tc: 530173961.5847317  
 115 196.631 93Nb(meta 1): 40771.640492079096  
 116 196.631 98Tc: 2366306.0908124447  
 117 196.631 97Mo: 172514065.24175954  
 118 196.631 96Mo: 26186498.548734967  
 119 196.631  
 120 196.631  
 121 196.631 Source isotope: 98Mo (42098)  
 122 196.631  
 123 196.631  
 124 196.631 Source isotope: 100Mo (42100)  
 125 196.631  
 126 216.077 Gamma Dose - Beam End  
 127 216.077 ======  
 128 216.077 Activity (Bq) 1.2188e+09  
 129 216.077 Gamma Energy (eV) 7.0916e+14  
 130 216.077 Gamma Energy (mW) 1.1362e-01  
 131 216.077 Absorbed Dose (mGy/s) 1.1299e-10  
 132 216.077 Absorbed Dose (mGy/hr) 4.0676e-07  
 133 216.077  
 134 216.077  
 135 216.077 Gamma Dose - Sim End  
 136 216.077 ======  
 137 216.077 Activity (Bq) 6.0485e+06  
 138 216.077 Gamma Energy (eV) 1.4080e+13  
 139 216.077 Gamma Energy (mW) 2.2558e-03  
 140 216.077 Absorbed Dose (mGy/s) 2.2433e-12  
 141 216.077 Absorbed Dose (mGy/hr) 8.0757e-09  
 142 216.077  
 143 216.077  
 144 216.077 Absorbed Dose Calculations  
 145 216.077 ======  
 146 216.077 Absorbed dose assumptions:  
 147 216.077 1. radiation from point, emitted isotropically  
 148 216.077 2. 80Kg human  
 149 216.077 3. 1m from point source  
 150 216.077 4. 1m squared surface area  
 151 216.077 5. all energy absorbed  
 152 216.077  
 153 216.077  
 154 216.077 Dose Limits  
 155 216.077 ======  
 156 216.077 employees 18+ 20 millisieverts/year  
 157 216.077 trainees 18+ 6 millisieverts/year  
 158 216.077 public and under 18s 1 millisievert/year  
 159 216.077 public and under 18s 1.140771128E-04 millisieverts/hour  
 160 216.077 public and under 18s 3.2E-08 millisieverts/second

161 216.077  
162 216.077 Dose averaged over area of skin not exceeding 1cm<sup>2</sup>  
163 216.077 Source: <http://www.hse.gov.uk/radiation/ionising/doses/>  
164 216.077  
165 216.077 1mW gammas completely absorbed by 80kg human will give public limit for entire year over 80 seconds.  
166 216.077  
167 216.077 Dose under 1W gammas absorbed for 1s no immediate changes (under 100 Rads)  
168 216.077 Dose 1W-2W gammas absorbed for 1s acute radiation syndrome (100-200 Rads)  
169 216.077 Dose 10W+ gammas absorbed for 1s mostly fatal (1000+ Rads)

---

### E.2.3 Total Decay Rates Over Time

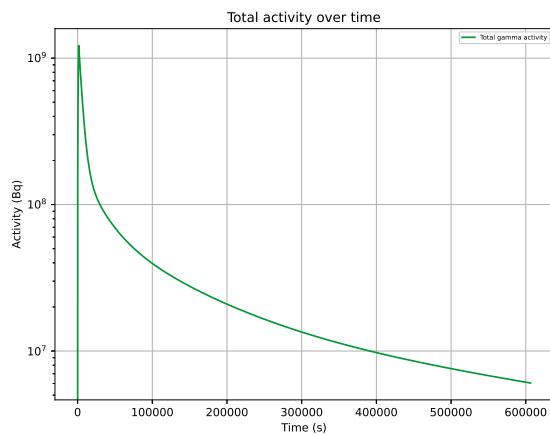


Figure E.25: Total activity over time

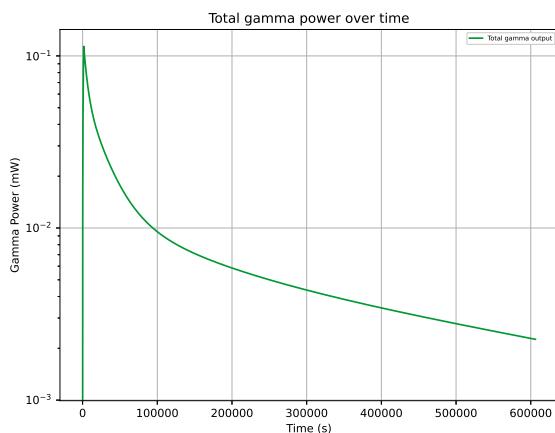


Figure E.26: Total gamma power output over time

#### E.2.4 Residual Activity Tallied by Target Isotope

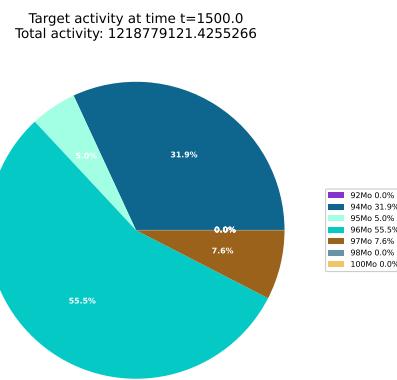


Figure E.27: Radioactive decay from residual isotopes tallied by target isotope - end of beam (1,500 seconds)

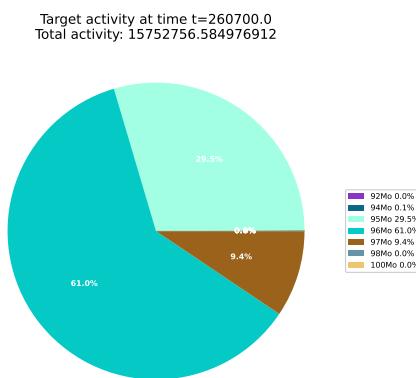


Figure E.28: Radioactive decay from residual isotopes tallied by target isotope - 3 days after irradiation

### E.2.5 Residual Activity Tallied by Residual Isotope

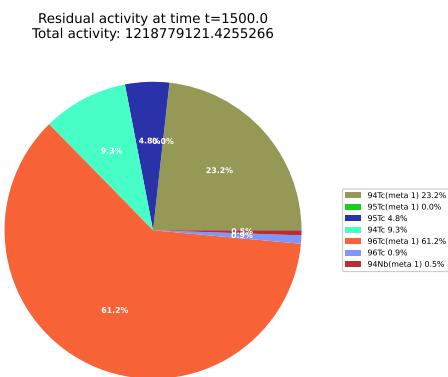


Figure E.29: Radioactive decay from residual isotopes tallied by residual isotope - end of beam (1,500 seconds)

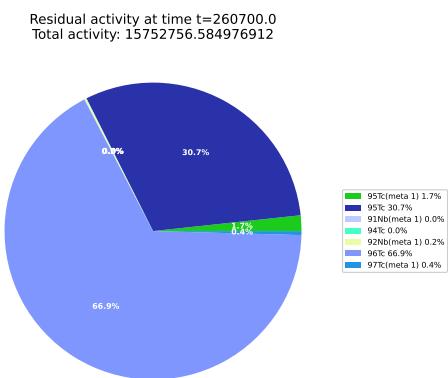


Figure E.30: Radioactive decay from residual isotopes tallied by residual isotope - 3 days after irradiation

### E.2.6 Residual Gamma Energy Tallied by Residual Isotope

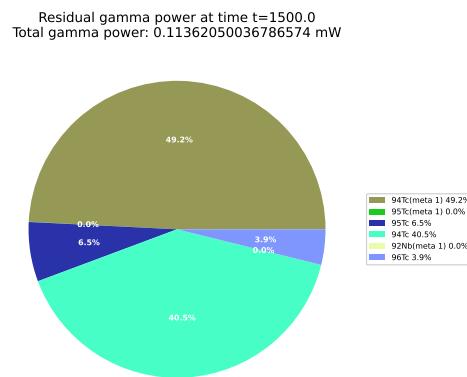


Figure E.31: Radioactive decay from residual isotopes tallied by residual isotope - end of beam (1,500 seconds)

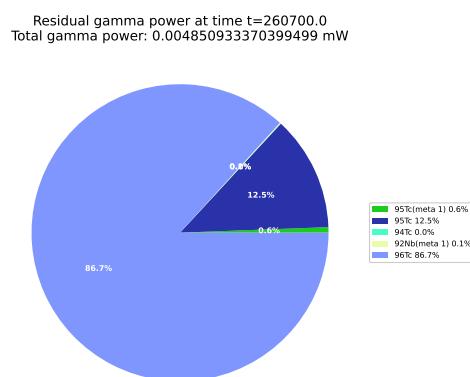


Figure E.32: Radioactive decay from residual isotopes tallied by residual isotope - 3 days after irradiation

### E.2.7 Gamma Lines

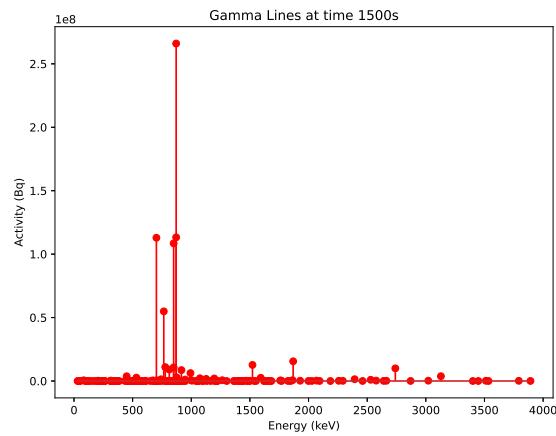


Figure E.33: Predicted gamma lines - end of beam (1,500 seconds)

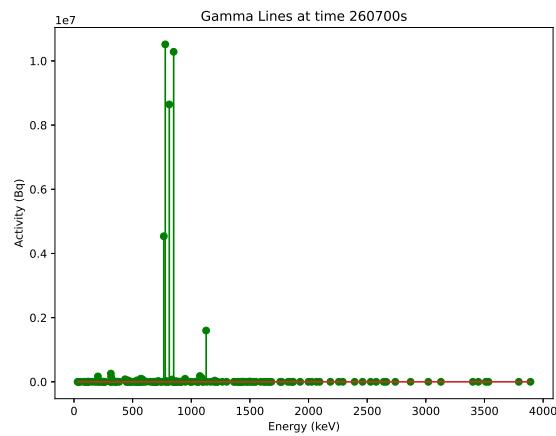


Figure E.34: Predicted gamma lines - 3 days after irradiation

## E.3 13MeV Proton Irradiated Molybdenum

### E.3.1 Calculation Settings

Material	Steel
Compotision	Fe: 96.375%, C,0.772%, Cu: 0.024%, Mn: 1.36%,
Ni: 0.698%, Si: 0.381%, Cr: 0.092%, V: 0.008%, P:0.009%, Si: 0.003%, Mo: 0.278%	
Sheet thickness	0.1mm
Density	10,220 $kgm^3$
Beam energy	13MeV
Beam projectile	proton
Beam current	5.0 microamps
Beam duration	1500s
Simulation end time	260,000s

Table E.3

## Appendix F

# Iron Activity: Irradiated to 100DPA

### F.1 Activity vs Beam Energy

The beam energy was set to 5MeV, 10MeV, 15MeV, 20MeV, 25MeV and 30MeV within SRIM, and the Activity V2 code was used to calculate how radioactive each sample would be if irradiated to 100 DPA of damage.

The beam settings are:

- target material = pure iron
- target thickness = 0.5mm
- beam current = 60 micro amps (maximum proton current for the Scanditronix MC-40)
- beam area =  $64\text{mm}^2$

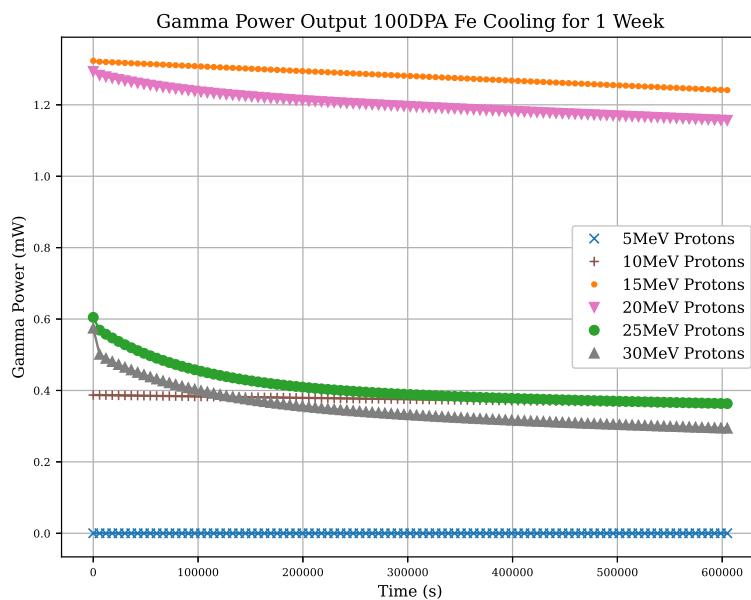


Figure F.1

### F.1.1 5MeV Proton Beam - 60 microamp 100DPA

With a proton energy of 5MeV the target must be irradiated for 205 days to reach 100DPA.

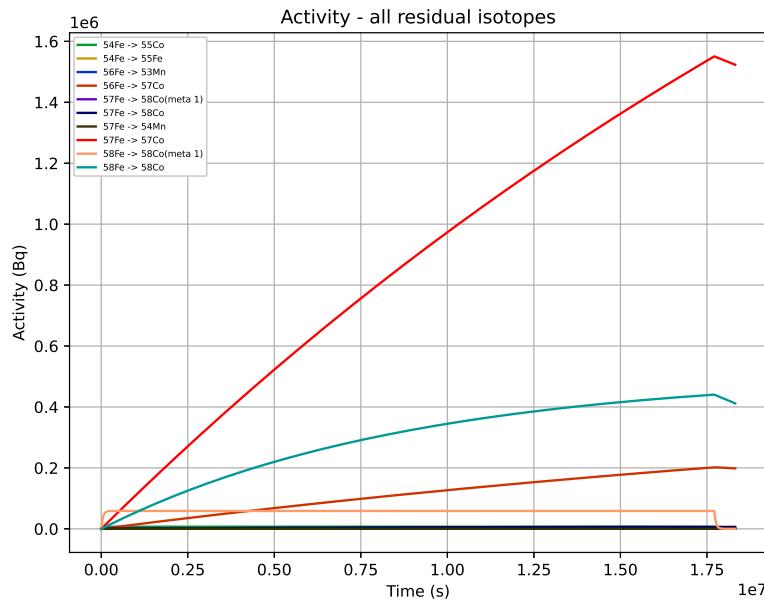


Figure F.2

Residual gamma power at time t=17712000.0  
Total gamma power: 9.435311052144465e-05 mW

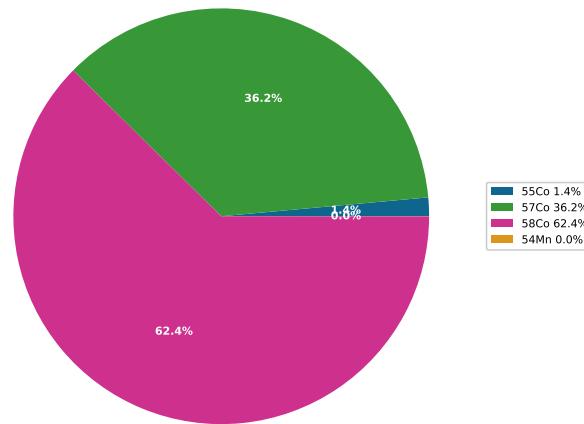


Figure F.3

Residual gamma power at time t=18316800.0  
Total gamma power: 8.852972608256112e-05 mW

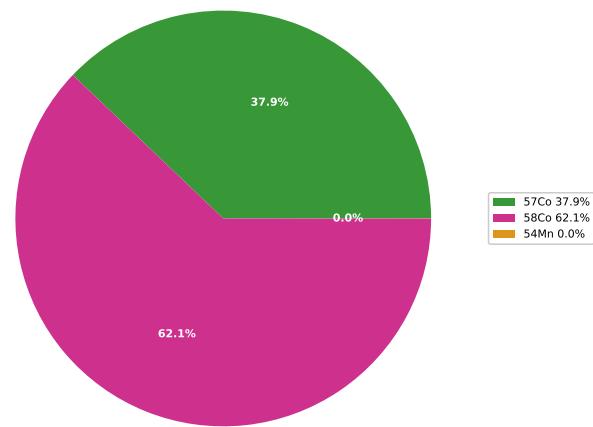


Figure F.4

### F.1.2 10MeV Proton Beam - 60 microamp 100DPA

The vacancies per ion are at a maximum out of the six settings here with a proton energy of 10MeV. As a result, the target must be irradiated for just 134 days to reach 100DPA.

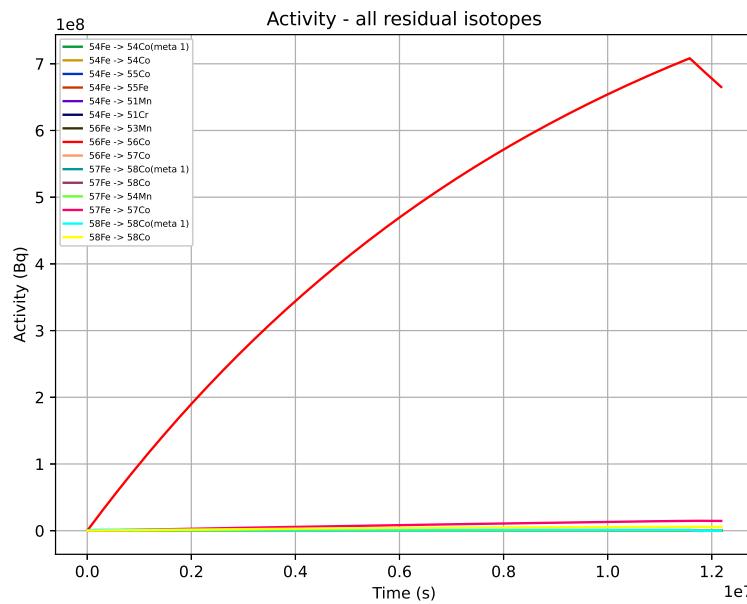


Figure F.5

Residual gamma power at time t=11577600.0  
Total gamma power: 0.3872607000597164 mW

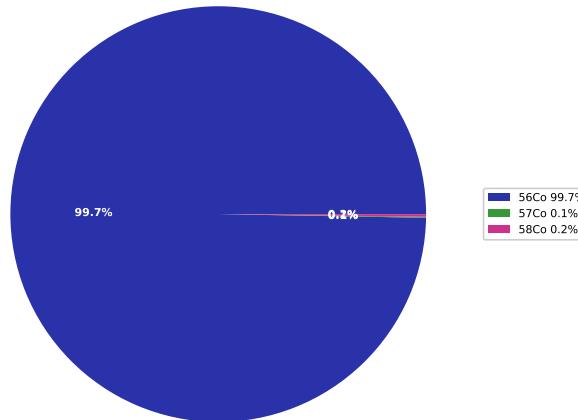


Figure F.6

Residual gamma power at time t=12182400.0  
Total gamma power: 0.3636776414065111 mW

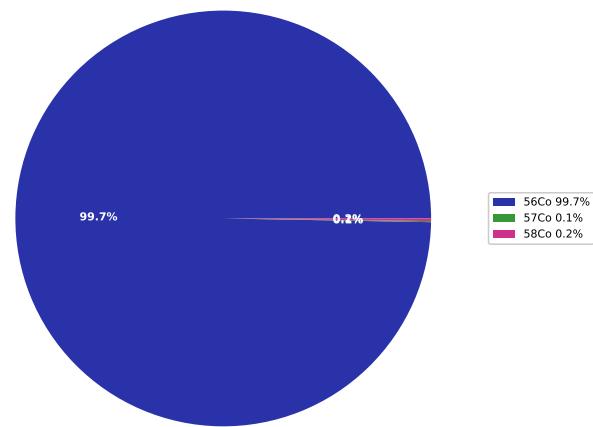


Figure F.7

### F.1.3 15MeV Proton Beam - 60 microamp 100DPA

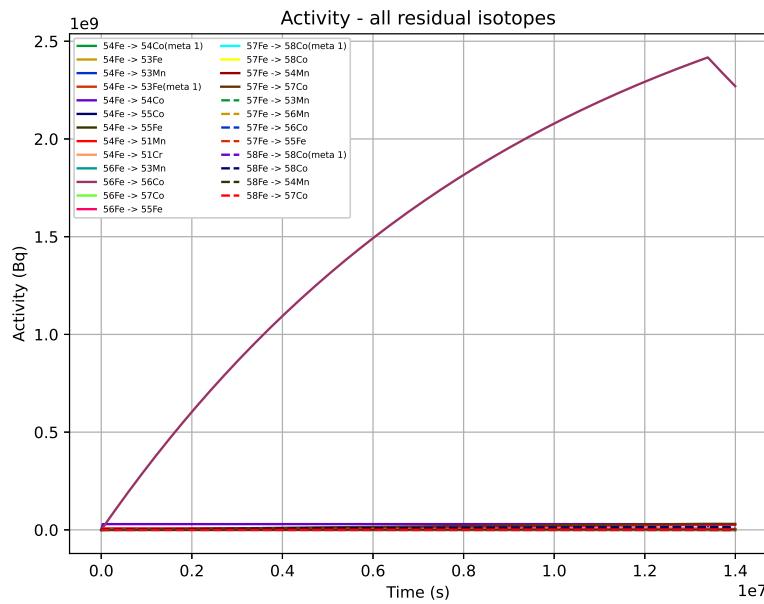


Figure F.8

Residual gamma power at time t=13392000.0  
Total gamma power: 1.3232690955956106 mW

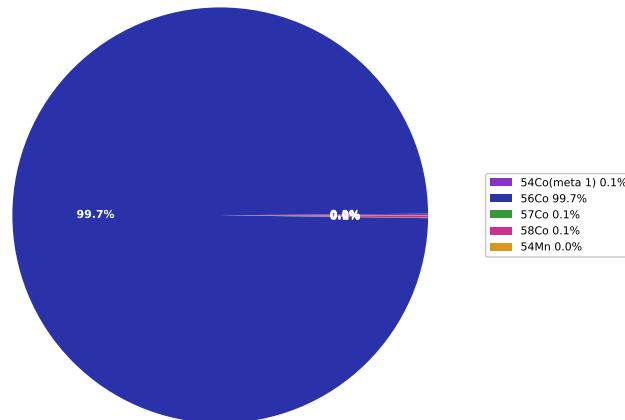


Figure F.9

Residual gamma power at time t=13996800.0  
Total gamma power: 1.2411065809416568 mW

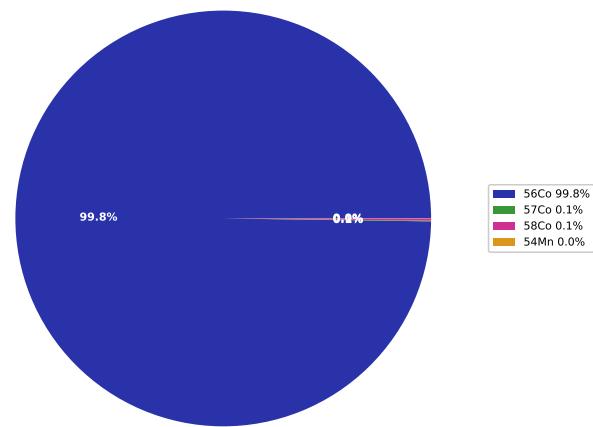


Figure F.10

### F.1.4 20MeV Proton Beam - 60 microamp 100DPA

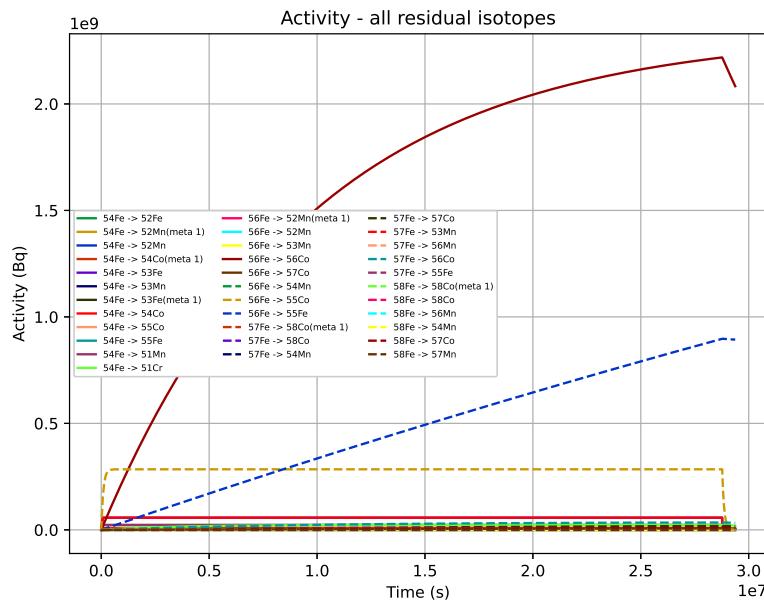


Figure F.11

Residual gamma power at time t=28771200.0  
Total gamma power: 1.2934181154506263 mW

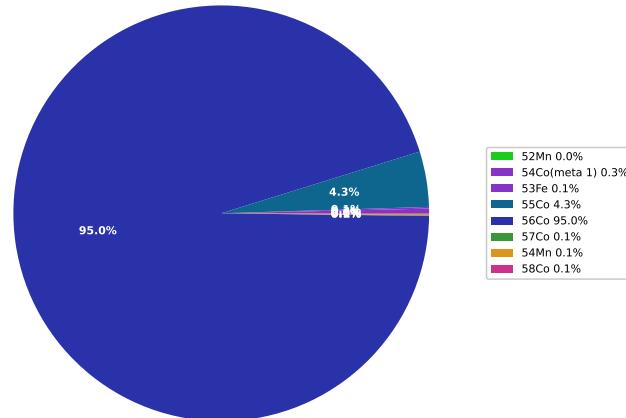


Figure F.12

Residual gamma power at time t=29376000.0  
Total gamma power: 1.1564199406474385 mW

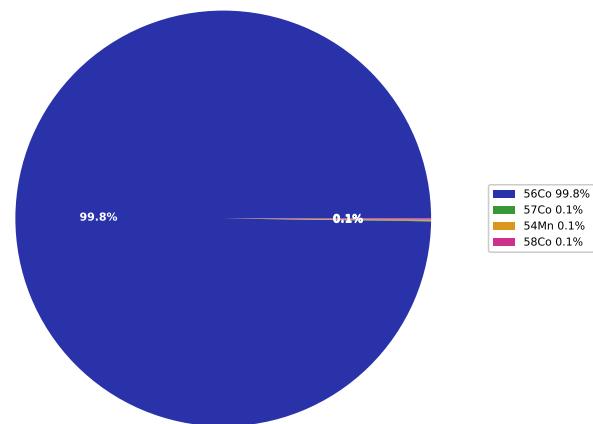


Figure F.13

### F.1.5 25MeV Proton Beam - 60 microamp 100DPA

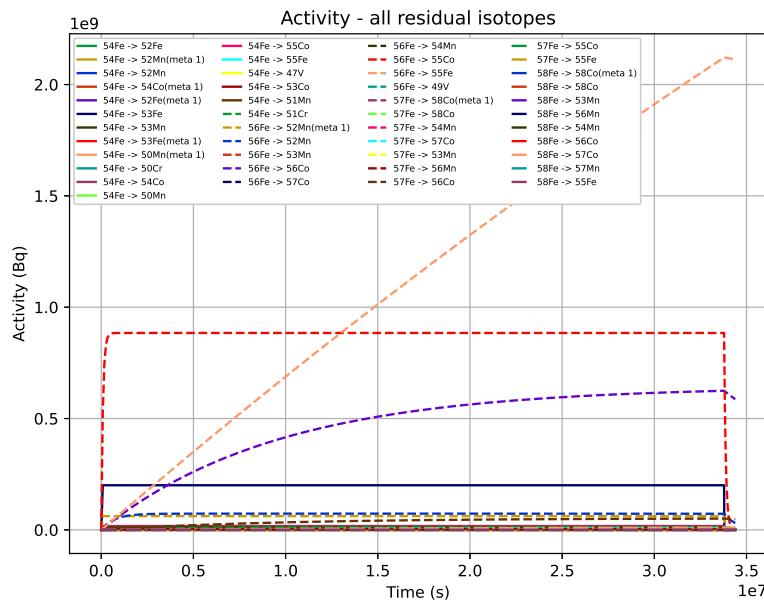


Figure F.14

Residual gamma power at time  $t=33782400.0$   
Total gamma power: 0.6047827408353267 mW

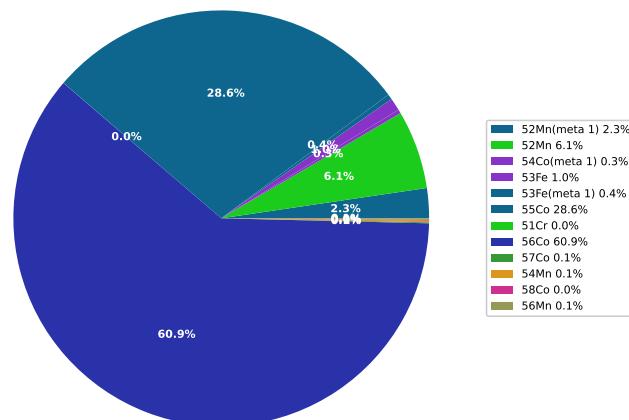


Figure F.15

Residual gamma power at time t=34387200.0  
Total gamma power: 0.363158186267968 mW

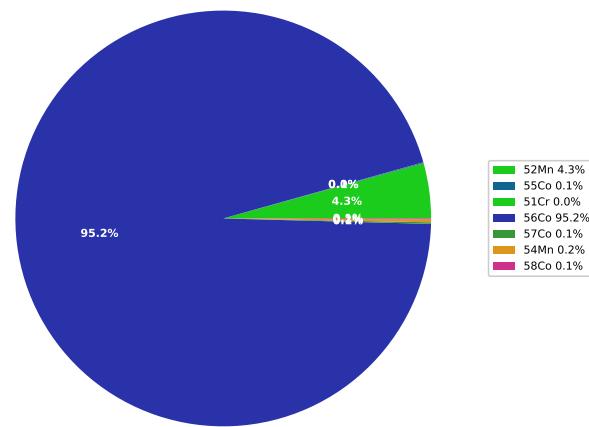


Figure F.16

#### F.1.6 30MeV Proton Beam - 60 microamp 100DPA

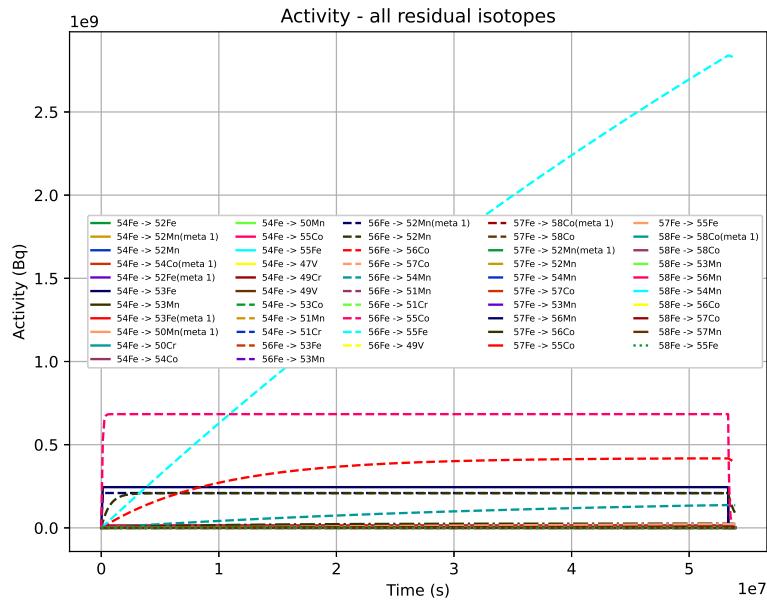


Figure F.17

Residual gamma power at time t=53312850.0  
Total gamma power: 0.5744878670833383 mW

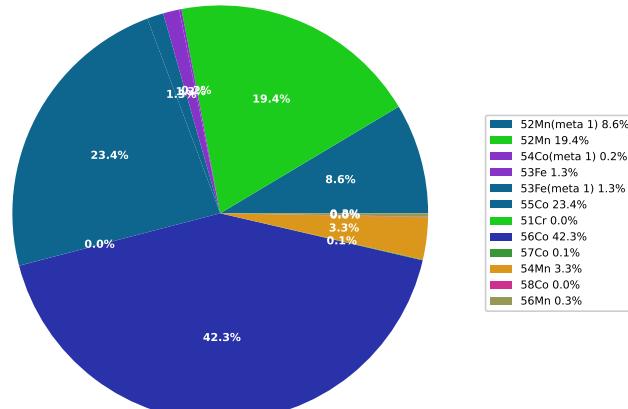


Figure F.18

Residual gamma power at time t=53917650.0  
Total gamma power: 0.2943024889030366 mW

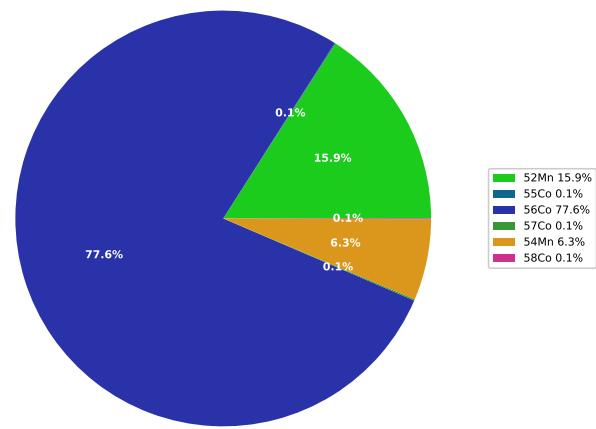


Figure F.19

# Appendix G

## SRIM Results

### G.0.1 Ion paths through an iron sample

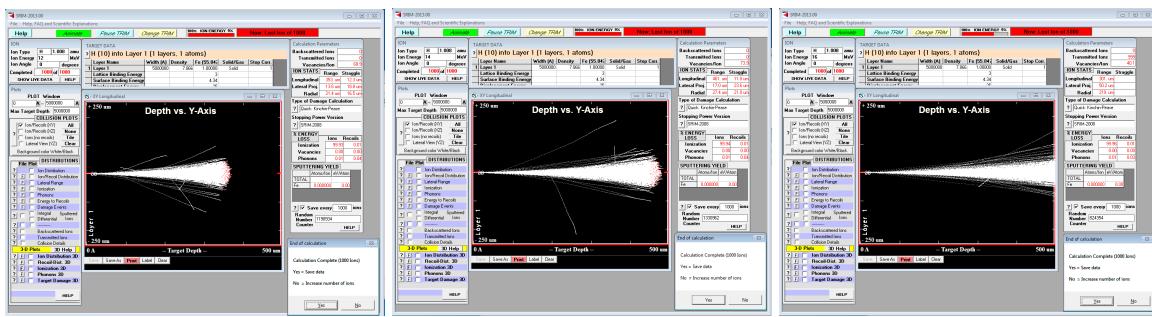


Figure G.1: Genetic Algorithm Population Size 512 vs Simulated Annealing

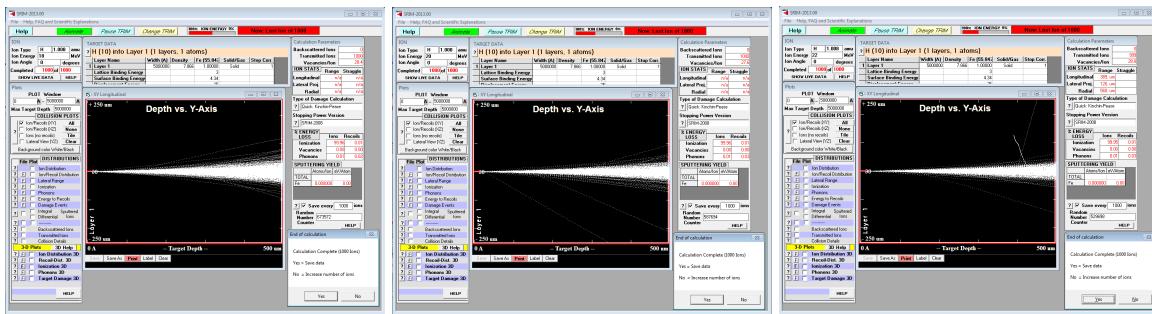


Figure G.2: Genetic Algorithm Population Size 512 vs Simulated Annealing

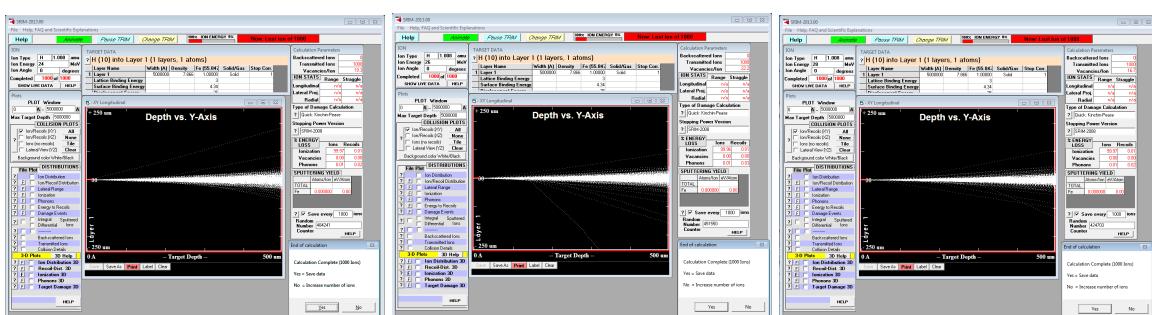


Figure G.3: Genetic Algorithm Population Size 512 vs Simulated Annealing

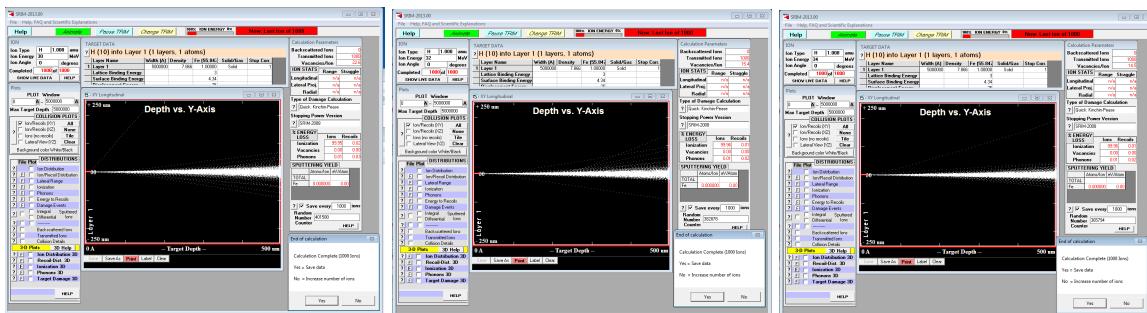


Figure G.4: Genetic Algorithm Population Size 512 vs Simulated Annealing

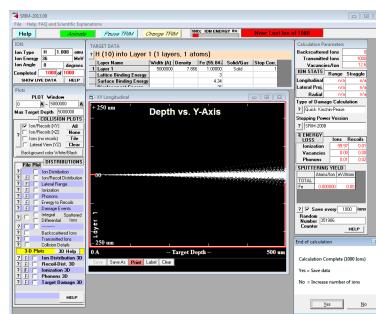


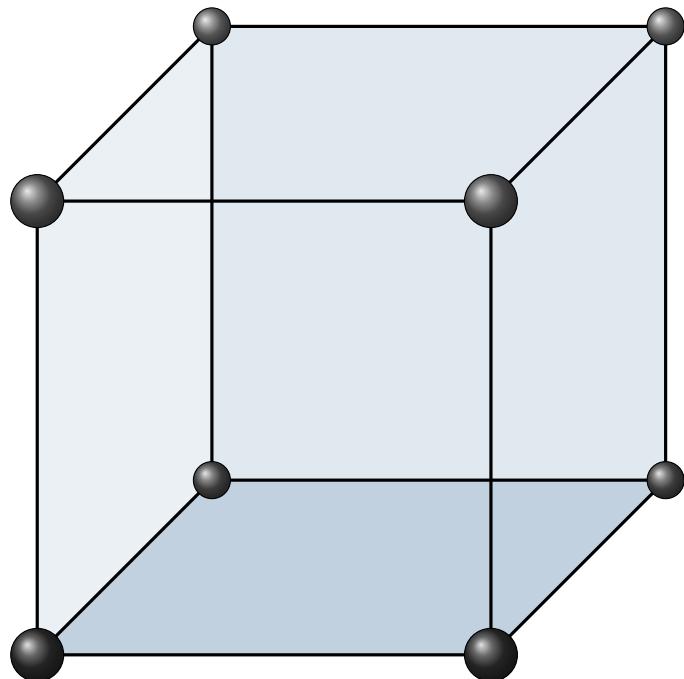
Figure G.5: Genetic Algorithm Population Size 512 vs Simulated Annealing

# Appendix H

## Crystal Structures

### H.1 Common Cubic Structures

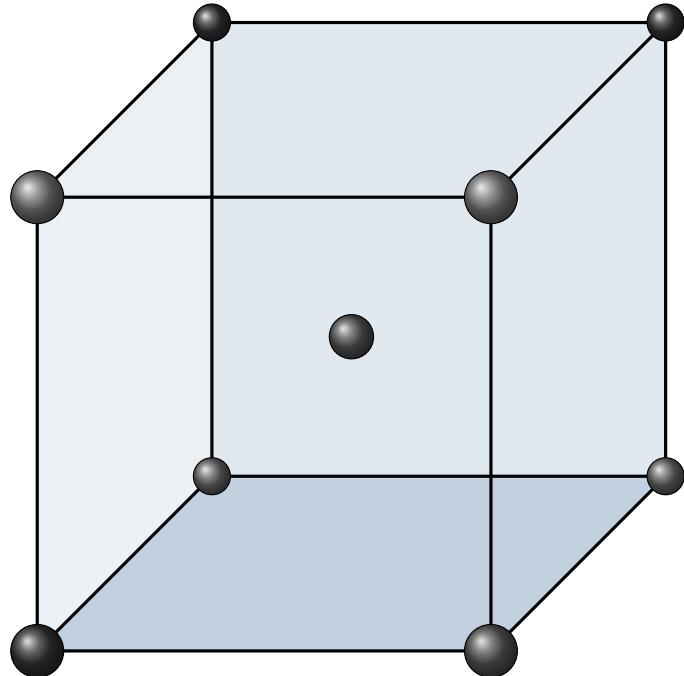
#### H.1.1 Simple Cubic



Sample Configuration  
[0.0 0.0 0.0]

Table H.1

### H.1.2 Body Centered Cubic

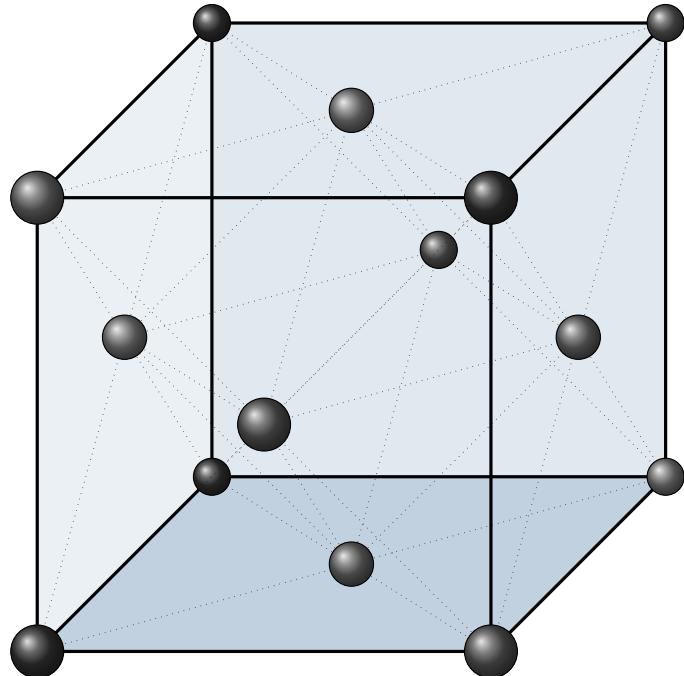


Sample Configuration

[0.0	0.0	0.0]
[0.5	0.5	0.5]

Table H.2

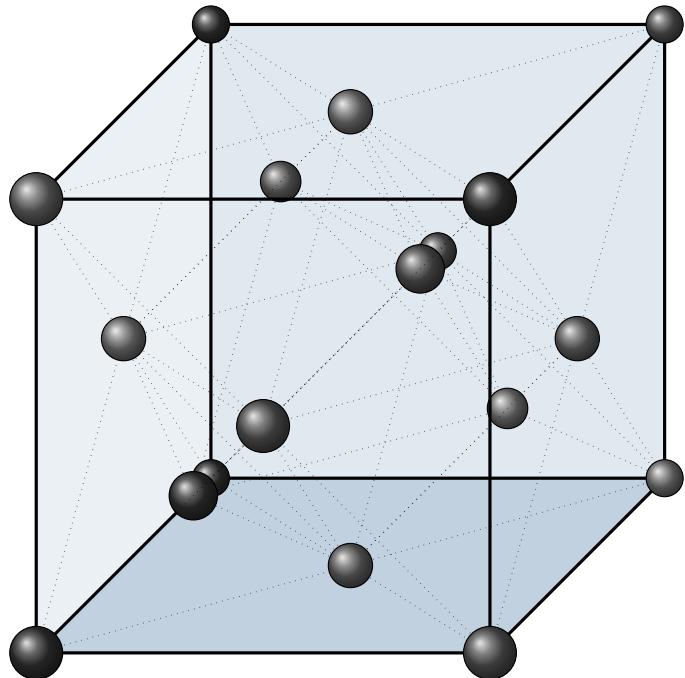
### H.1.3 Face Centered Cubic



Sample Configuration

[0.0	0.0	0.0]
[0.0	0.5	0.5]
[0.5	0.0	0.5]
[0.5	0.5	0.0]

Table H.3

**H.1.4 Zincblende**

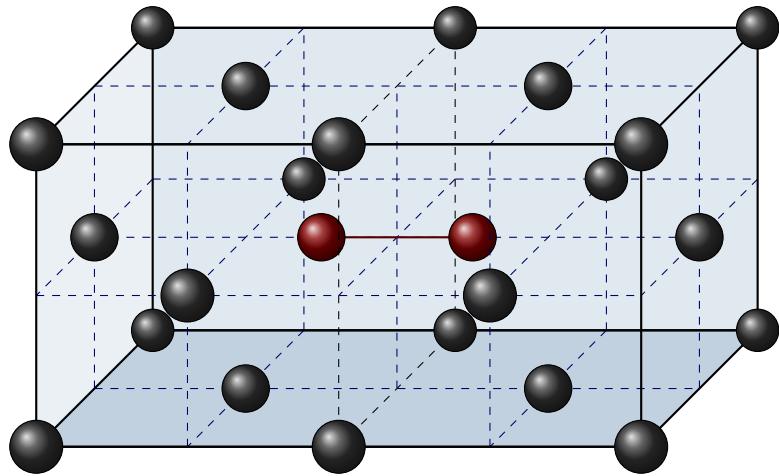
Sample Configuration

[0.0	0.0	0.0]
[0.0	0.5	0.5]
[0.5	0.0	0.5]
[0.5	0.5	0.0]
[0.25	0.25	0.25]
[0.75	0.75	0.25]
[0.25	0.75	0.75]
[0.75	0.25	0.75]

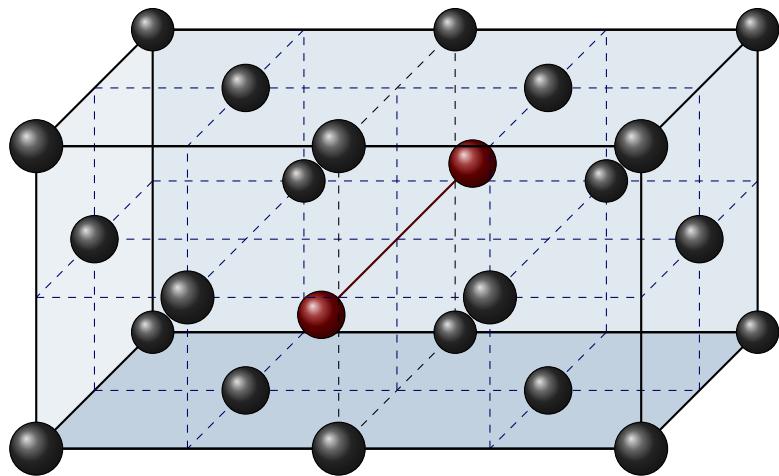
Table H.4

## H.2 FCC Defects

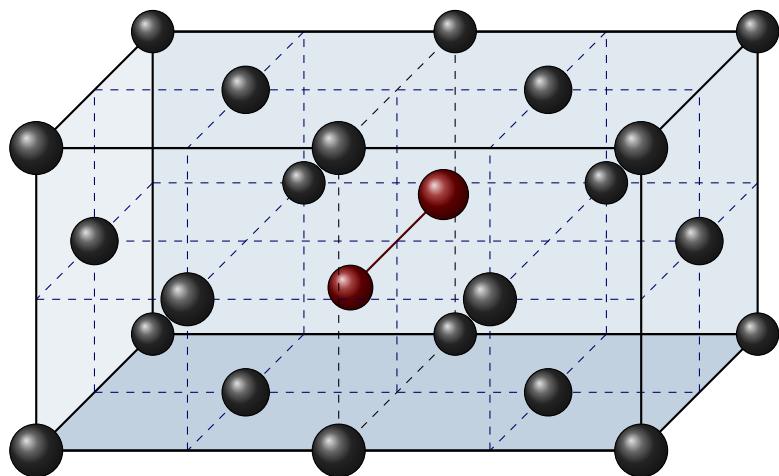
### H.2.1 Dumbbell $<100>$



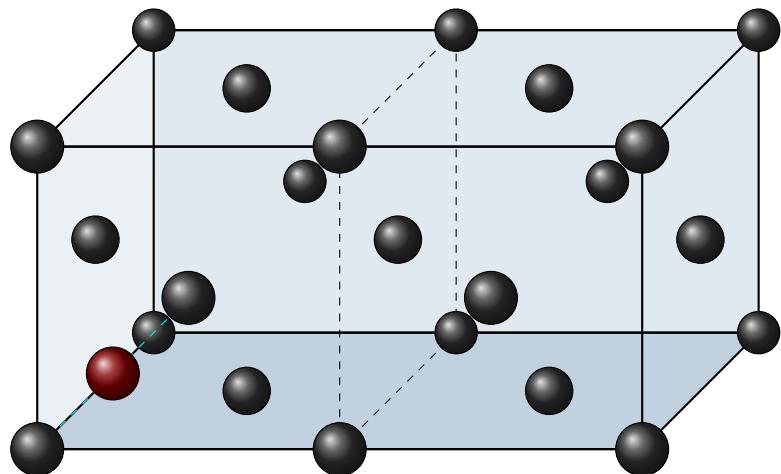
### H.2.2 Dumbbell $<110>$



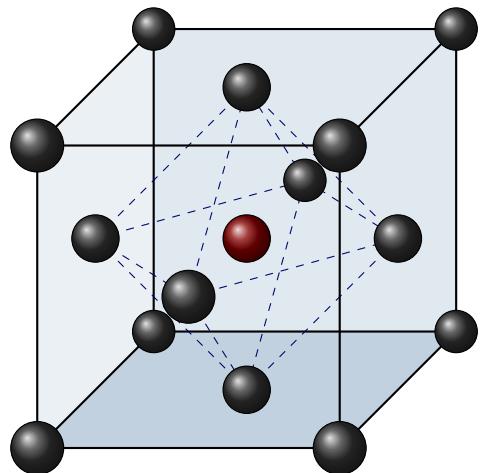
### H.2.3 Dumbbell $<111>$



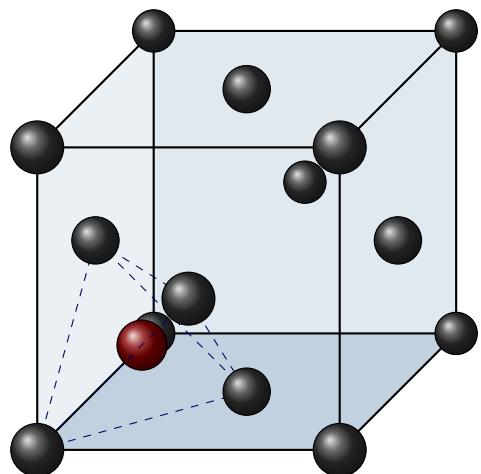
#### H.2.4 Crowdion



#### H.2.5 Octahedral Interstitial



#### H.2.6 Tetrahedral Interstitial



# Appendix I

## Elastic Constants

### I.1 Elastic Constant Matrices

Crystal:	Cubic
Sides:	$a = b = c$
Angles:	$\alpha = \beta = \gamma = \frac{\pi}{2}$ radians
Symmetry:	$\frac{\pi}{2}$ radians rotation, 3 mutually orthogonal planes of symmetry
No. Independent Elastic Constants:	3
Elastic Constants:	$\begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{bmatrix}$

Table I.1

Crystal:	Orthorhombic/Orthotropic
Sides:	$a \neq b \neq c$
Angles:	$\alpha = \beta = \gamma = \frac{\pi}{2}$ radians
Symmetry:	3 mutually orthogonal planes of symmetry
No. Independent Elastic Constants:	9
Elastic Constants:	$\begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix}$

Table I.2

Crystal:	Monoclinic
Sides:	$a \neq b \neq c$
Angles:	$\alpha < \frac{\pi}{2}$ radians, $\beta = \gamma = \frac{\pi}{2}$ radians
Symmetry:	symmetry in the xy plane
No. Independent Elastic Constants:	13
Elastic Constants:	$\begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & C_{16} \\ C_{12} & C_{22} & C_{23} & 0 & 0 & C_{26} \\ C_{13} & C_{23} & C_{33} & 0 & 0 & C_{36} \\ 0 & 0 & 0 & C_{44} & C_{45} & 0 \\ 0 & 0 & 0 & C_{45} & C_{55} & 0 \\ C_{16} & C_{26} & C_{36} & 0 & 0 & C_{66} \end{bmatrix}$

Table I.3

Crystal:	Triclinic
Sides:	$a \neq b \neq c$
Angles:	$\alpha < \frac{\pi}{2}$ radians, $\beta < \frac{\pi}{2}$ radians, $\gamma < \frac{\pi}{2}$ radians
Symmetry:	no symmetry
No. Independent Elastic Constants:	21
Elastic Constants:	$\begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & C_{16} \\ C_{12} & C_{22} & C_{23} & 0 & 0 & C_{26} \\ C_{13} & C_{23} & C_{33} & 0 & 0 & C_{36} \\ 0 & 0 & 0 & C_{44} & C_{45} & 0 \\ 0 & 0 & 0 & C_{45} & C_{55} & 0 \\ C_{16} & C_{26} & C_{36} & 0 & 0 & C_{66} \end{bmatrix}$

Table I.4

## I.2 Sets of Strains Applied to Calculate Elastic Constants

Strains are applied to a relaxed cell to determine the elastic constants. Various methods have been used, and these differ from paper to paper based on the crystal structure and the author. Several of these strains have been used throughout this work in various computer codes, including the QE\_EoS and EAMPA codes.

### I.2.1 Cubic Crystals - Mehl, Singh, Klein and Papaconstantopoulos 1993

$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$E(\sigma)$
$\sigma$	$\sigma$	$\sigma$	0	0	0	BM EoS Fit $B_0$
$\sigma$	$-\sigma$	$\frac{\sigma^2}{1-\sigma^2}$	0	0	0	$\Delta E(\sigma) = \Delta E(-\sigma) = V_0((C_{11} - C_{12})\frac{\sigma}{2} + O[\sigma^4])$
0	0	$\frac{\sigma^2}{4-\sigma^2}$	0	0	$\sigma$	$\Delta E(\sigma) = \Delta E(-\sigma) = V_0 C_{44} \frac{\sigma}{2} + O[\sigma^4]$

Table I.5: Calculation of elastic constants using the equation of state, a volume conserving orthorhombic strain and volume conserving monoclinic strain.[106][100]

These values are inserted into the strain matrix in eq. ??.

$$\vec{\epsilon} = \begin{bmatrix} e_1 & \frac{e_6}{2} & \frac{e_5}{2} \\ \frac{e_6}{2} & e_2 & \frac{e_4}{2} \\ \frac{e_5}{2} & \frac{e_4}{2} & e_3 \end{bmatrix} \quad (\text{I.1})$$

The values for  $C_{11}$  and  $C_{12}$  are computed using the polynomial fit to the change in energy due to the second strain and the bulk modulus.

$$\begin{bmatrix} 1 & -1 \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix} \begin{bmatrix} C_{11} \\ C_{12} \end{bmatrix} = \begin{bmatrix} \frac{C_p}{V_0} \\ B_0 \end{bmatrix} \quad (\text{I.2})$$

### I.2.2 Orthorhombic Crystals - Ravindran, Fast, Korzhavyi and Johansson

$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$E(\sigma)$
$\sigma$	0	0	0	0	0	$E(V, \sigma) = E(V_0, 0) + V_0 (\tau_1 \sigma + \frac{C_{11}}{2} \sigma^2)$
0	$\sigma$	0	0	0	0	$E(V, \sigma) = E(V_0, 0) + V_0 (\tau_2 \sigma + \frac{C_{22}}{2} \sigma^2)$
0	0	$\sigma$	0	0	0	$E(V, \sigma) = E(V_0, 0) + V_0 (\tau_3 \sigma + \frac{C_{33}}{2} \sigma^2)$
$\frac{1}{(1-\sigma^2)^{\frac{1}{3}}}$	$\frac{1}{(1-\sigma^2)^{\frac{1}{3}}}$	$\frac{1}{(1-\sigma^2)^{\frac{1}{3}}}$	$\frac{1}{(1-\sigma^2)^{\frac{1}{3}}}$	0	0	$E(V, \sigma) = E(V_0, 0) + V_0 (2\tau_4 \sigma + 2C_{44} \sigma^2)$
$\frac{1}{(1-\sigma^2)^{\frac{1}{3}}}$	$\frac{1}{(1-\sigma^2)^{\frac{1}{3}}}$	$\frac{1}{(1-\sigma^2)^{\frac{1}{3}}}$	0	$\frac{1}{(1-\sigma^2)^{\frac{1}{3}}}$	0	$E(V, \sigma) = E(V_0, 0) + V_0 (2\tau_5 \sigma + 2C_{55} \sigma^2)$
$\frac{1}{(1-\sigma^2)^{\frac{1}{3}}}$	$\frac{1}{(1-\sigma^2)^{\frac{1}{3}}}$	$\frac{1}{(1-\sigma^2)^{\frac{1}{3}}}$	0	0	$\frac{1}{(1-\sigma^2)^{\frac{1}{3}}}$	$E(V, \sigma) = E(V_0, 0) + V_0 (2\tau_6 \sigma + 2C_{66} \sigma^2)$
$\frac{1+\sigma}{(1-\sigma^2)^{\frac{1}{3}}}$	$\frac{1-\sigma}{(1-\sigma^2)^{\frac{1}{3}}}$	$\frac{1}{(1-\sigma^2)^{\frac{1}{3}}}$	0	0	0	$E(V, \sigma) = E(V_0, 0) + V_0 ((\tau_1 - \tau_2) \sigma + \frac{C_{11} + C_{22} - 2C_{12}}{2} \sigma^2)$
$\frac{1+\sigma}{(1-\sigma^2)^{\frac{1}{3}}}$	$\frac{1}{(1-\sigma^2)^{\frac{1}{3}}}$	$\frac{1-\sigma}{(1-\sigma^2)^{\frac{1}{3}}}$	0	0	0	$E(V, \sigma) = E(V_0, 0) + V_0 ((\tau_1 - \tau_3) \sigma + \frac{C_{11} + C_{33} - 2C_{13}}{2} \sigma^2)$
$\frac{1}{(1-\sigma^2)^{\frac{1}{3}}}$	$\frac{1+\sigma}{(1-\sigma^2)^{\frac{1}{3}}}$	$\frac{1-\sigma}{(1-\sigma^2)^{\frac{1}{3}}}$	0	0	0	$E(V, \sigma) = E(V_0, 0) + V_0 ((\tau_2 - \tau_3) \sigma + \frac{C_{22} + C_{33} - 2C_{23}}{2} \sigma^2)$

Table I.6: Calculation of orthorhombic elastic constants by Ravindran et al.[dfttisiravindran]

These values are inserted into the strain matrix in eq. ??.

$$\vec{\epsilon} = \begin{bmatrix} e_1 & e_6 & e_5 \\ e_6 & e_2 & e_4 \\ e_5 & e_4 & e_3 \end{bmatrix} \quad (\text{I.3})$$

The 9 elastic constants are calculated by polynomial fit to the strain-energy data. Using the second order of each polynomial,  $p_1$  to  $p_9$ , the elastic constants are calculated by solving eq ??.

$$\begin{bmatrix} \frac{V_0}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{V_0}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{V_0}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2V_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2V_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2V_0 & 0 & 0 & 0 \\ \frac{V_0}{2} & \frac{V_0}{2} & 0 & 0 & 0 & 0 & -V_0 & 0 & 0 \\ \frac{V_0}{2} & 0 & \frac{V_0}{2} & 0 & 0 & 0 & 0 & -V_0 & 0 \\ 0 & \frac{V_0}{2} & \frac{V_0}{2} & 0 & 0 & 0 & 0 & 0 & -V_0 \end{bmatrix} \begin{bmatrix} C_{11} \\ C_{22} \\ C_{33} \\ C_{44} \\ C_{55} \\ C_{66} \\ C_{12} \\ C_{13} \\ C_{23} \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \\ p_9 \end{bmatrix} \quad (I.4)$$

### I.2.3 Cubic Crystals - Connetable and Thomas 2009

$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$E(\sigma)$
$\sigma$	$\sigma$	$\sigma$	0	0	0	BM EoS Fit $B_0$
$\sigma$	0	0	0	0	0	$\Delta E(\sigma) = \frac{V_0}{2}C_{11}\sigma^2 + O[\sigma^4]$
0	0	0	$2\sigma$	$2\sigma$	$2\sigma$	$\Delta E(\sigma) = 6V_0C_{44}\sigma^2 + O[\sigma^4]$

Table I.7: Calculation of elastic constants for a cubic crystal[137]

$$\vec{\epsilon} = \begin{bmatrix} 2e_1 & e_6 & e_5 \\ e_6 & 2e_2 & e_4 \\ e_5 & e_4 & 2e_3 \end{bmatrix} \quad (I.5)$$

### I.2.4 Orthorhombic Crystals - Connetable and Thomas 2009

$$\vec{\epsilon} = \begin{bmatrix} 2e_1 & e_6 & e_5 \\ e_6 & 2e_2 & e_4 \\ e_5 & e_4 & 2e_3 \end{bmatrix} \quad (I.6)$$

$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$E(\sigma)$
$\sigma$	0	0	0	0	0	$\Delta E(\sigma) = \frac{V_0}{2} C_{11} \sigma^2$
0	$\sigma$	0	0	0	0	$\Delta E(\sigma) = \frac{V_0}{2} C_{22} \sigma^2$
0	0	$\sigma$	0	0	0	$\Delta E(\sigma) = \frac{V_0}{2} C_{33} \sigma^2$
0	0	0	$2\sigma$	0	0	$\Delta E(\sigma) = 2V_0 C_{44} \sigma^2$
0	0	0	0	$2\sigma$	0	$\Delta E(\sigma) = 2V_0 C_{55} \sigma^2$
0	0	0	0	0	$2\sigma$	$\Delta E(\sigma) = 2V_0 C_{66} \sigma^2$
$u(1+\sigma) - 1$	$u(1-\sigma) - 1$	$u - 1$	0	0	0	$\Delta E(\sigma) = \frac{V_0}{2} (C_{11} + C_{22} - 2C_{12}) \sigma^2$
$u(1+\sigma) - 1$	$u - 1$	$u(1-\sigma) - 1$	0	0	0	$\Delta E(\sigma) = \frac{V_0}{2} (C_{11} + C_{33} - 2C_{13}) \sigma^2$
$u - 1$	$u(1+\sigma) - 1$	$u(1-\sigma) - 1$	0	0	0	$\Delta E(\sigma) = \frac{V_0}{2} (C_{22} + C_{33} - 2C_{23}) \sigma^2$

Table I.8: Calculation of elastic constants for a cubic crystal[137],  $u = (1 - \sigma^2)^{-\frac{1}{3}}$

## Appendix J

# PBE Functional

The GGA PBE functional was the primary type used in this work for the DFT calculated data. The LDA, LSDA and GGA are described in detail in “Density functionals from LDA to GGA”[150].

The local density approximation is:

$$E_{xc}^{LDA}[\rho] = \int d^3r \rho(\vec{r}) [e_x(\rho(\vec{r})) + e_c(\rho(\vec{r}))] \quad (\text{J.1})$$

Exchange energy per electron of the unpolarized uniform electron gas:

$$e_x(\rho) \quad (\text{J.2})$$

Correlation energy per electron of the unpolarized uniform electron gas:

$$e_c(\rho) \quad (\text{J.3})$$

Splitting the density into spin up and spin down:

$$\rho_{\uparrow}(\vec{r}) = \sum_k^{occ} |\phi_{k,\uparrow}(\vec{r})|^2 \quad \rho_{\downarrow}(\vec{r}) = \sum_k^{occ} |\phi_{k,\downarrow}(\vec{r})|^2 \quad \rho(\vec{r}) = \rho_{\uparrow}(\vec{r}) + \rho_{\downarrow}(\vec{r}) \quad (\text{J.4})$$

The relative spin polarization is:

$$\zeta = \frac{\rho_{\uparrow} - \rho_{\downarrow}}{\rho_{\uparrow} + \rho_{\downarrow}} \quad (\text{J.5})$$

The LSDA is:

$$E_{xc}^{LSDA}[\rho_{\uparrow}, \rho_{\downarrow}] = \int d^3r \rho(\vec{r}) [e_x(\rho(\vec{r})) f(\zeta(\vec{r})) + e_c(r_s(\vec{r}), \zeta(\vec{r}))] \quad (\text{J.6})$$

where

$$f(\zeta) = \frac{1}{2} \left[ (1 + \zeta)^{\frac{4}{3}} + (1 - \zeta)^{\frac{4}{3}} \right] \quad (\text{J.7})$$

The correlation energy per electron of the spin-polarized uniform electron gas is:

$$e_c(r_s, \zeta) \quad (\text{J.8})$$

where  $r_s$  is the local Seitz radius:

$$\begin{aligned} \rho &= 3/4\pi r_s^3 \\ r_s &= \sqrt[3]{\frac{4\rho}{3\pi}} \end{aligned} \quad (\text{J.9})$$

The PBE correlation energy functional is:

$$E_c^{PBE}[\rho_\uparrow, \rho_\downarrow] = \int d^3r \rho [e_c(r_s, \zeta) + H(r_s, \zeta, t)] \quad (\text{J.10})$$

where

$$t = \left( \frac{\pi}{4} \right)^{\frac{1}{2}} \left( \frac{9\pi}{4} \right)^{\frac{1}{6}} \frac{s}{\phi r_s^{\frac{1}{2}}} \quad (\text{J.11})$$

$$s = \frac{|\nabla \rho|}{2(3\pi^2)^{\frac{1}{3}} \rho^{\frac{4}{3}}} = \frac{3}{2} \left( \frac{4}{9\pi} \right)^{\frac{1}{3}} a_B |\nabla r_s| \quad (\text{J.12})$$

$$a_B = \frac{\hbar^2}{m\epsilon^2} \text{ (the Bohr radius)} \quad (\text{J.13})$$

$$\phi(\zeta) = \frac{1}{2} \left[ (1 + \zeta)^{\frac{2}{3}} + (1 - \zeta)^{\frac{2}{3}} \right] \quad (\text{J.14})$$

$$H = \gamma \phi^3 \ln \left[ 1 + \frac{\beta}{\gamma} t^2 \left( \frac{1 + At^2}{1 + At^2 + A^2 t^4} \right) \right] \quad (\text{J.15})$$

$$A = \left( \frac{\beta}{\gamma} \right) \frac{1}{\exp(-e_c(r_s, \zeta)/(\gamma \phi^3)) - 1} \quad (\text{J.16})$$

where  $\beta = 0.066725$  and  $\gamma = 0.031091$  Hartree.

The PBE exchange energy functional is:

$$E_x^{PBE}[\rho] = \int d^3r \rho e_x(\rho) F_x(s) \quad (\text{J.17})$$

where

$$F_x(s) = 1 + \kappa - \frac{\kappa}{1 + \mu s^2 / \kappa} \quad (\text{J.18})$$

where  $\kappa = 0.804$  and  $\mu = 0.21951$ .

The spin density functional is:

$$E_x^{PBE}[\rho_\uparrow, \rho_\downarrow] = \frac{1}{2} E_x^{PBE}[2\rho_\uparrow] + \frac{1}{2} E_x^{PBE}[2\rho_\downarrow] \quad (\text{J.19})$$

## Appendix K

### DFT Convergence Plots

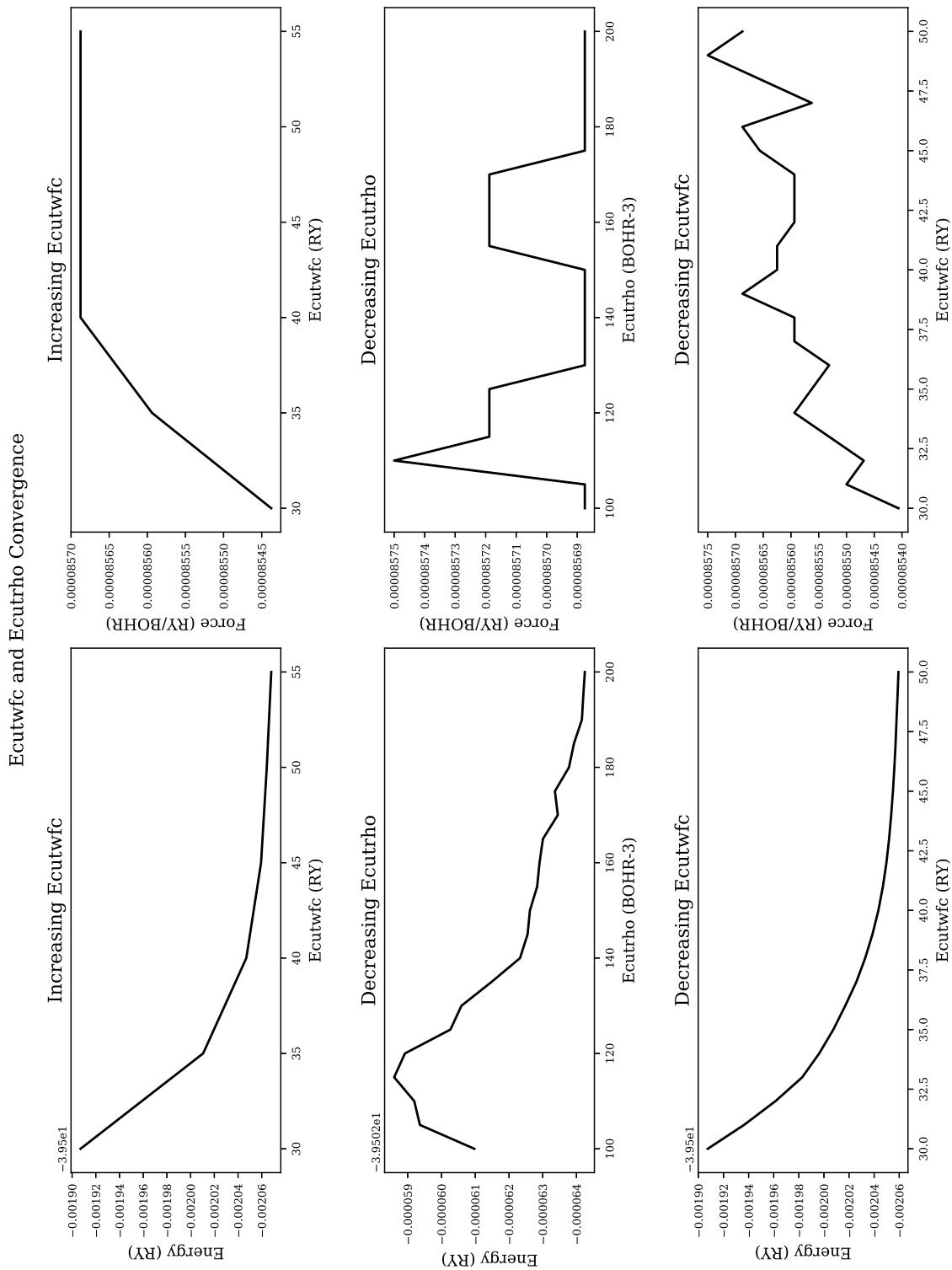


Figure K.1:  $E_{\text{cutwfc}}$  and  $E_{\text{cutrho}}$  convergence for aluminium (energy and force)

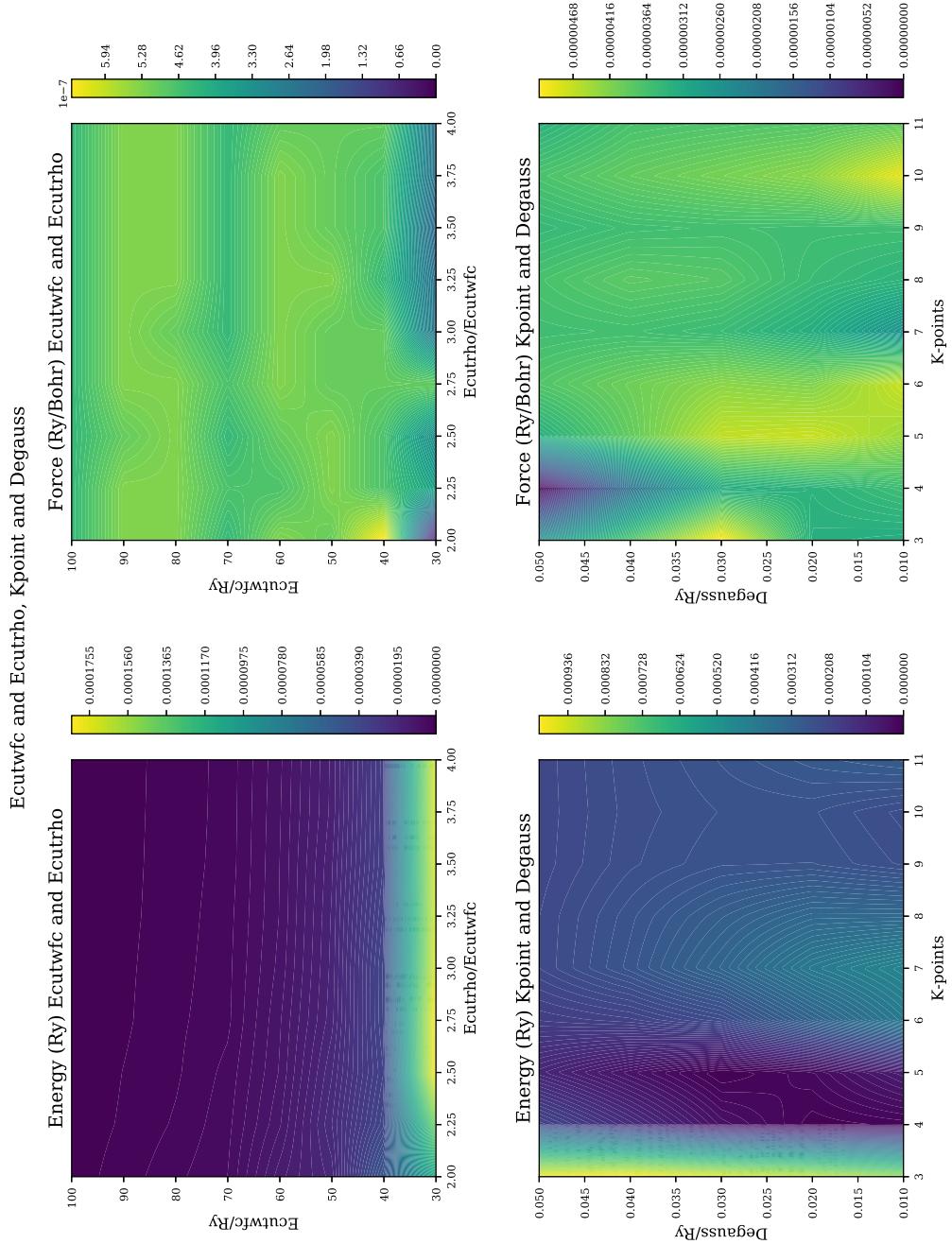


Figure K.2: 2D density map - force and energy convergence with k-points and smearing - alluminium

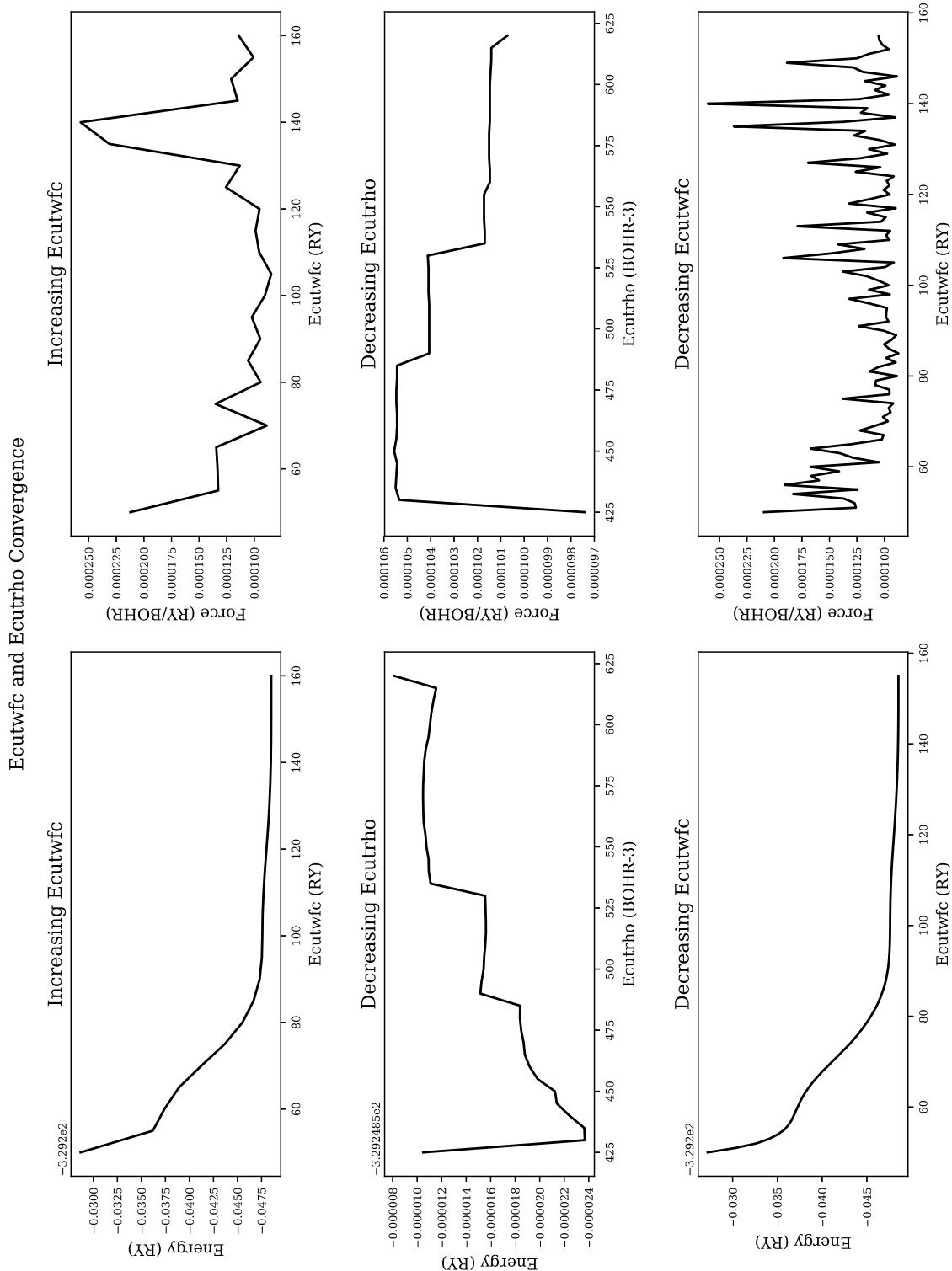


Figure K.3: Ecutwfc and Ecutfro convergence for iron (energy and force)

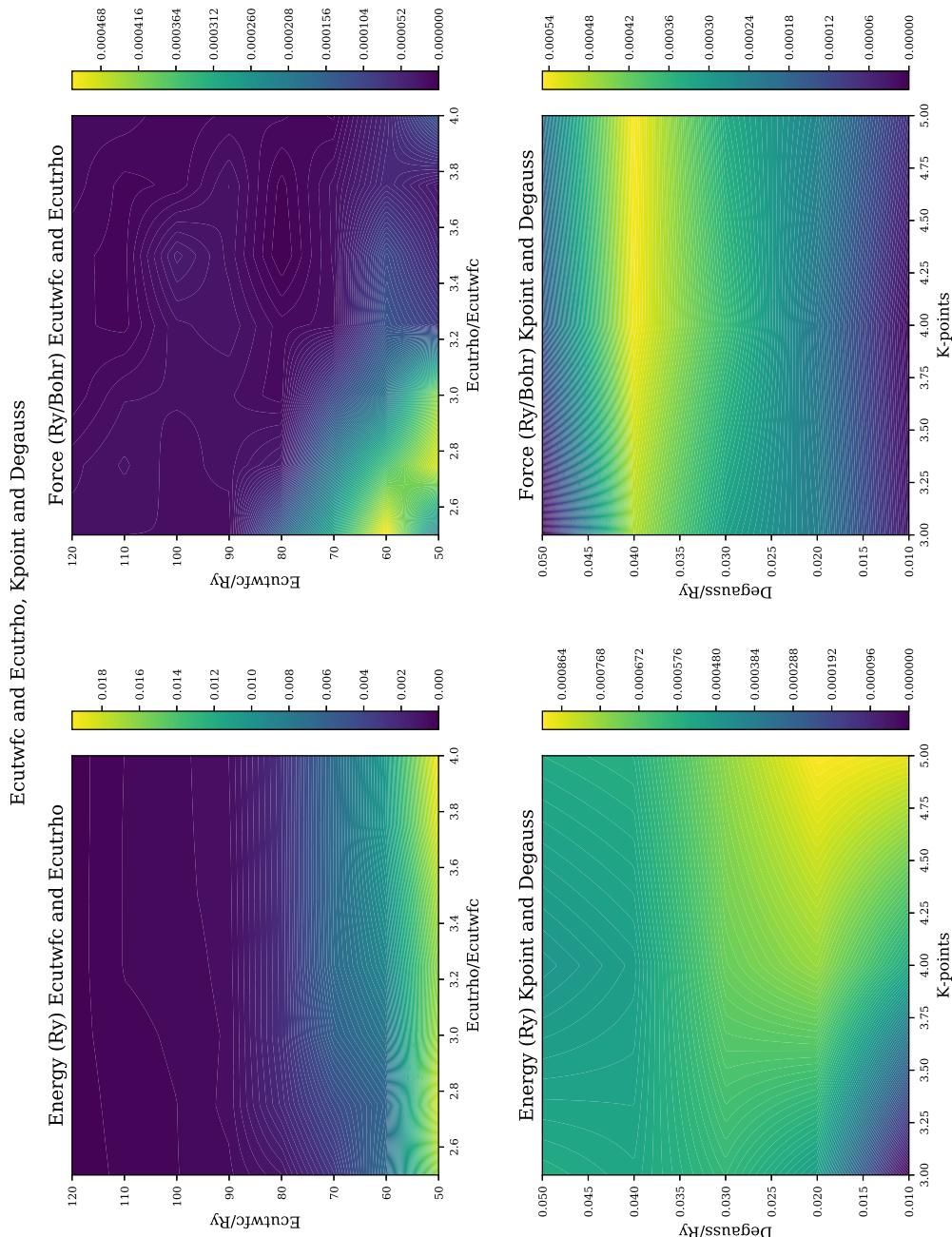


Figure K.4: 2D density map - force and energy convergence with k-points and smearing - iron

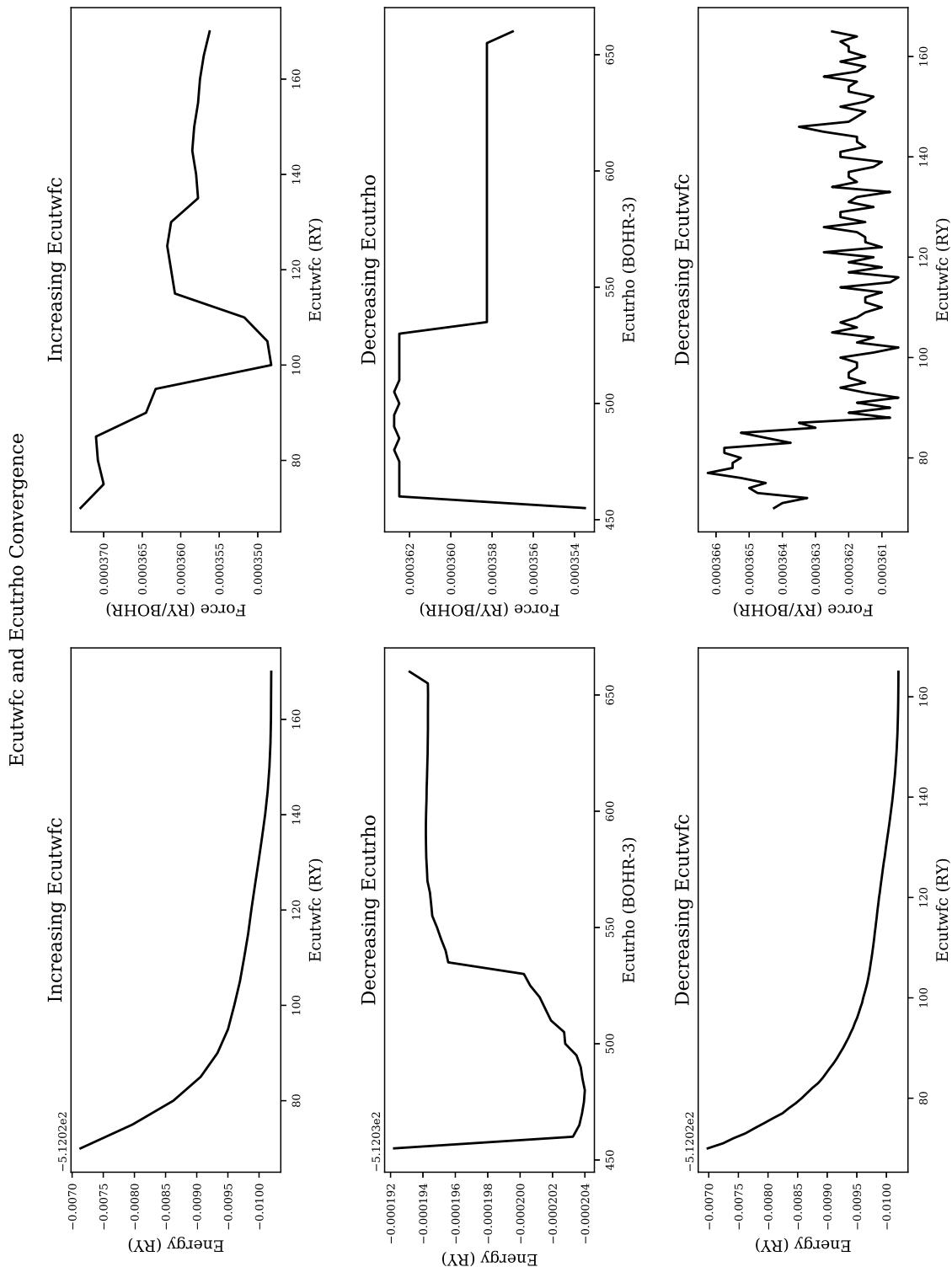


Figure K.5: Ecutwfc and Ecutfc convergence for palladium (energy and force)

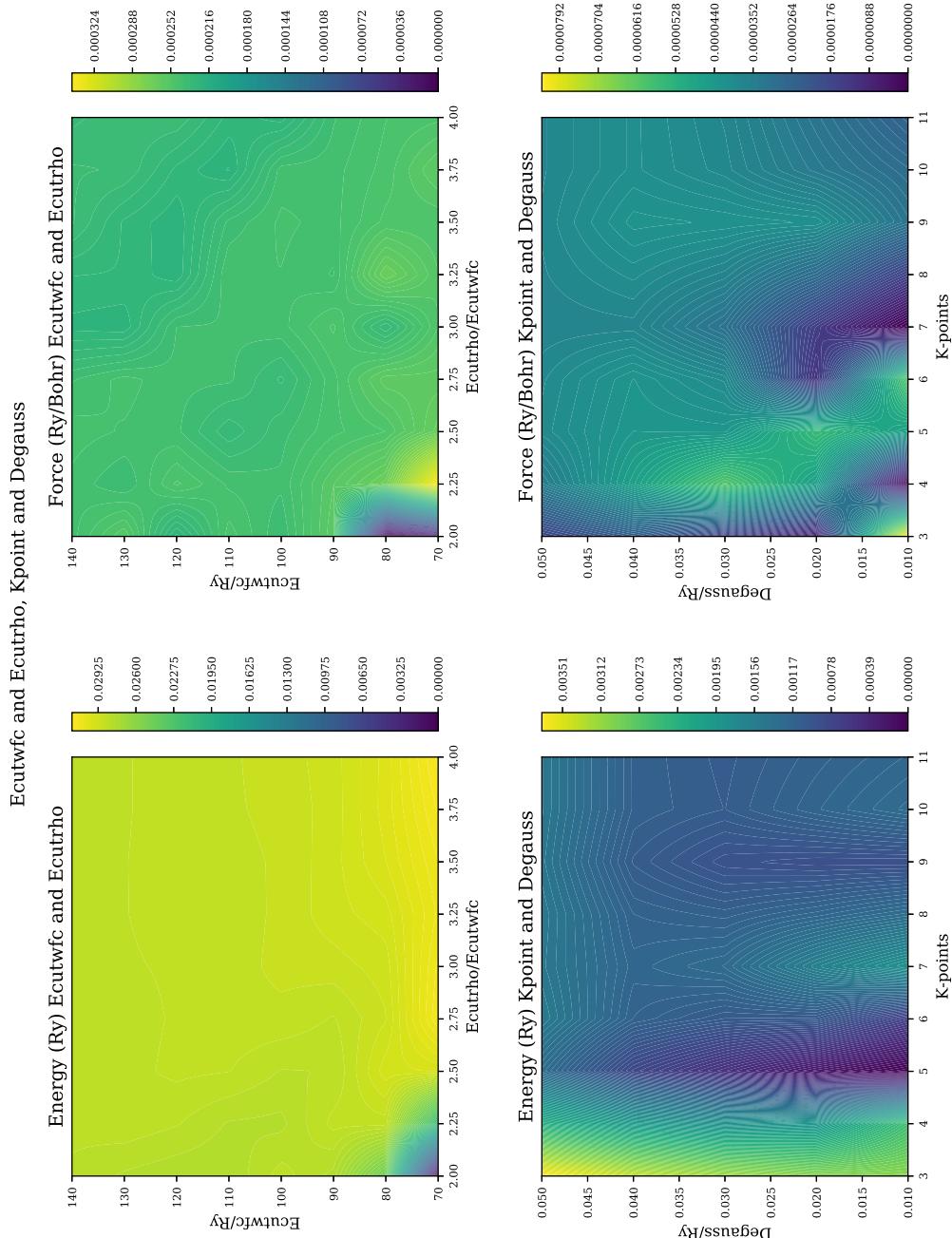


Figure K.6: 2D density map - force and energy convergence with k-points and smearing - palladium

## **Appendix L**

### **QEEOS**

## Appendix M

# DFT Calculated Properties

### M.1 QEEOS

These results were calculated using the QEEOS python program that automatically creates and runs PWscf input files, collects the output files and processes the data.

## M.2 FCC Aluminium

### M.2.1 DFT Settings for PWscf

The calculations were performed with the Quantum Espresso PWscf package along with the QEEOS python program.

Setting	Value
ecutwfc (Ry)	50
ecutrho	200
smearing (Ry)	0.04
k-points	11 11 11 1 1 1
nspin	0 (Non-magnetic)
pseudopotential	Al.pbe-nl-kjpaw_psl.1.0.0.UPF
etot_conv_thr	0.0001
forc_conv_thr	0.001
conv_thr	1.0D-6
diagonalization	david
mixing_beta	0.1
mixing_mode	plain

Table M.1: Aluminium DFT settings

### M.2.2 Resulting Plots

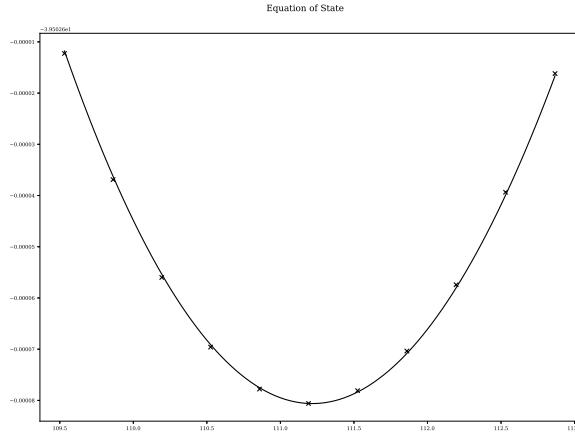


Figure M.1: Aluminium FCC equation of state plot

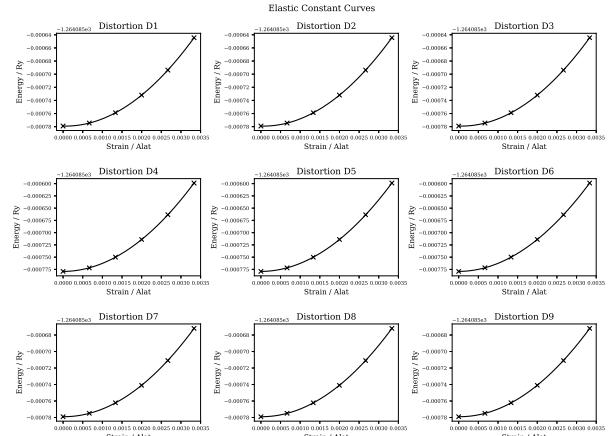


Figure M.2: Aluminium FCC elastic constants plot

### M.2.3 Aluminium FCC Results File

Listing M.1: Aluminium FCC DFT Calculated Properties

---

```

1 ######
2 #          RESULTS          #
3 #####
4 Thu, 26 Mar 2020 13:32:45 +0000
5
6 INPUT SETTINGS
7
8 Structure:           fcc
9 Size:                2
10 Alat (Bohr):        6.9
11 Alat (Bohr) (exp.): 13.8
12 Alat (Ang):         3.6501000000000006
13 Alat (Ang) (exp.):  7.300200000000001
14 Atoms in crystal:   32
15 Atoms in crystal (exp.): 32
16 AMU per crystal:    863.423999999994
17
18 Ecutfwfc:           50
19 Ecutrho:             200
20 Degauss:             0.04
21 Kpoints:             automatic 11 11 11 1 1 1
22
23 EOS Points:          11
24 Total Strain:         0.005
25
26 V0 (Bohr^3):         111.192295309
27 E0:                  -39.5027342866
28 B0 (RY/BOHR3):       0.00531912635008
29 B0 (GPA):             77.572208693
30 BOP:                 2.0
31
32 EC Points:            9
33 Total Strain:          0.005
34
35
36 Relaxed alat (Bohr):  7.6325799
37 Relaxed alat (Bohr) (exp): 15.2651598
38 Relaxed alat (Ang):     4.0376347671
39 Relaxed alat (Ang) (exp): 8.0752695342
40 Relaxed cp:             1.0   0.0   0.0
41                      0.0   1.0   0.0
42                      0.0   0.0   1.0
43 Relaxed volume (Bohr^3): 3557.16544569
44 Relaxed density (kgm^-3): 2722.71416748
45
46 Stiffness (RY/BOHR3):  0.0076028 0.0039542 0.0039563 0.0   0.0   0.0
47                      0.0039542 0.0075973 0.003945 0.0   0.0   0.0
48                      0.0039563 0.003945 0.0076068 0.0   0.0   0.0
49                      0.0   0.0   0.0   0.0023326 0.0   0.0
50                      0.0   0.0   0.0   0.0   0.0023349 0.0
51                      0.0   0.0   0.0   0.0   0.0   0.0023339
52
53 Stiffness (GPA):      110.877131 57.6666837 57.6978857 0.0   0.0   0.0
54                      57.6666837 110.796070 57.5331112 0.0   0.0   0.0
55                      57.6978857 57.5331112 110.934314 0.0   0.0   0.0
56                      0.0   0.0   0.0   34.0184756 0.0   0.0
57                      0.0   0.0   0.0   0.0   34.0513488 0.0
58                      0.0   0.0   0.0   0.0   0.0   34.037456

```

```

59
60 Compliance (1/GPA):      0.0140133 -0.0048022 -0.0047979 0.0     0.0     0.0
61                         -0.0048022  0.0139977 -0.0047619 0.0     0.0     0.0
62                         -0.0047979 -0.0047619  0.0139794 0.0     0.0     0.0
63                         0.0      -0.0     -0.0      0.0293958 -0.0    -0.0
64                         0.0      0.0      0.0      0.0      0.0293674 0.0
65                         0.0      0.0      0.0      0.0      0.0      0.0293794
66
67 Bulk Modulus B (GPA):    75.378064563
68 BR (GPA):                75.3780315039
69 BV (GPA):                75.378097622
70
71 Shear Modulus G (GPA):   30.8455908955
72 GR:                      30.6224032645
73 GV:                      31.0687785264
74
75 Young's Modulus E (GPA): 81.4294731854
76
77 Poisson's Ratio ν:       0.319953205977
78
79
80 L Elastic Wave V:       0.206857953612
81 T Elastic Wave V:       0.106437709845
82 M Elastic Wave V:       0.119193708075
83
84 Debye Temperature:      0.00139515659606
85
86 Melting Point:          1300.0K

```

---

#### M.2.4 Known and Calculated Values

	Al Experimental	Al DFT (this work)
Structure	Face Centered Cubic	Face Centered Cubic
$a_0$ (Angs)	4.05 Angstrom	4.04 Angstrom
Nearest Neighbour	2.86 Angstrom	2.86 Angstrom
Basis vectors	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$
$E_{coh}$ (eV)	-3.36 eV	Not Calculated
Bulk Modulus $B_0$ (GPA)	76	77.6
Bulk Modulus $B_{0,r}$ (GPA)	-	75.4
Bulk Modulus $B_{0,g}$ (GPA)	-	75.4
Young's modulus $E$ (GPA)	70	81.4
Shear Modulus $G$ (GPA)	26	30.8
Poisson Ratio $\nu$	0.35	0.32
Elastic Constants (GPA)	$\begin{bmatrix} 114 & 62 & 62 & 0 & 0 & 0 \\ 62 & 114 & 62 & 0 & 0 & 0 \\ 62 & 62 & 114 & 0 & 0 & 0 \\ 0 & 0 & 0 & 32 & 0 & 0 \\ 0 & 0 & 0 & 0 & 32 & 0 \\ 0 & 0 & 0 & 0 & 0 & 32 \end{bmatrix}$	$\begin{bmatrix} 110.9 & 57.7 & 57.7 & 0 & 0 & 0 \\ 57.7 & 110.8 & 57.5 & 0 & 0 & 0 \\ 57.7 & 57.5 & 110.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 34.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 34.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 34.0 \end{bmatrix}$

Table M.2: FCC Aluminium: Experimental Values vs DFT Calculated Variables

## M.3 BCC Iron [No Magnetism]

### M.3.1 Major DFT Settings

Ecutwfc: 71

Ecutrho: 430

Smearing: 0.04

K-points: 9 9 9 1 1 1

Nspin: 1 (non-polarized calculation)

Pseudopotential: Fe.pbe-spn-kjpaw-psl.1.0.0.UPF

### M.3.2 Equation of State and Elastic Constants

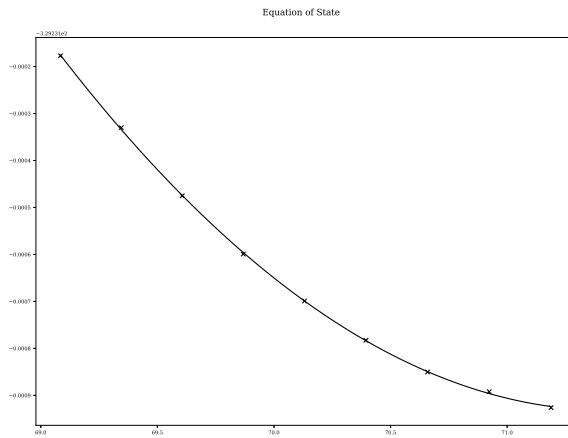


Figure M.3: Iron BCC equation of state plot

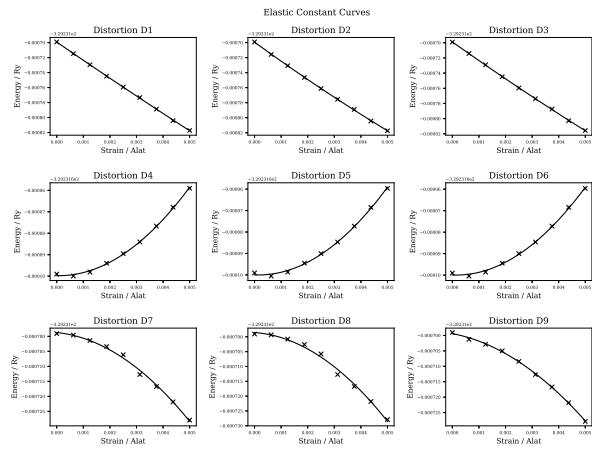


Figure M.4: Iron BCC elastic constants plot

### M.3.3 Iron BCC Results File

Listing M.2: Iron BCC DFT Calculated Properties

---

```

1 ######
2 #          RESULTS          #
3 #####
4 Tue, 03 Nov 2020 06:01:22 +0000
5
6 INPUT SETTINGS
7 #####
8
9 Structure:           bcc
10 Size:                2
11 AO:                  2.8
12 AO Units:            ang
13 AO (Bohr):          10.584
14 AO (Angs):           5.598936
15
16 Ecutwfc:             71
17 Ecutrho:              430
18 Degauss:              0.04
19 Kpoints:              9 9 9 1 1 1
20 Kpoints Type:         automatic
21 EOS Points:           9
22 EOS Strain:            0.005
23 EC Points:             9
24 EC Strain:              0.005
25
26
27 RELAXED
28 #####
29
30 a0 (Bohr):           10.391466456
31 CP:                  1.0    0.0    0.0
32                      0.0    1.0    0.0
33                      0.0    0.0    1.0
34 Volume (Bohr^3/unit cell): 1122.09730705
35 Density KG/m^3:        8932.14485799
36 AMU per crystal unit:  893.5200000000003
37 Atoms per crystal unit: 16
38 Total Energy/Ry:       -5267.70721439
39 Energy per Atom/Ry:    -329.231700899375
40 Total Force Ry/Bohr:   1.7e-05
41 Force per atom Ry/Bohr: 1.0625e-06
42
43
44 a0 primitive (Bohr):   5.195733228
45 a0 primitive (Ang):     2.748542877612
46
47
48 EOS
49 #####
50
51 V0 (Bohr^3 / atom):    71.4368199046
52 E0 (Ry / atom):         -329.231932186
53 B0 (RY/BOHR3):          0.0195284174344
54 B0 (GPA):               287.272784669
55 BOP:                   4.98933790197
56
57
58 EC

```

```

59 #####
60
61 Stiffness (RY/BOHR3):   0.0043229 0.0225755 0.0216226 0.0      0.0      0.0
62                      0.0225755 0.0121066 0.0217004 0.0      0.0      0.0
63                      0.0216226 0.0217004 0.0083439 0.0      0.0      0.0
64                      0.0      0.0      0.0      0.0121324 0.0      0.0
65                      0.0      0.0      0.0      0.0      0.0123943 0.0
66                      0.0      0.0      0.0      0.0      0.0      0.0123441
67
68 Stiffness (GPA):       63.04417 329.233427 315.336007 0.0      0.0      0.0
69                      329.233427 176.558337 316.470988 0.0      0.0      0.0
70                      315.336007 316.470988 121.684505 0.0      0.0      0.0
71                      0.0      0.0      0.0      176.934632 0.0      0.0
72                      0.0      0.0      0.0      0.0      180.753497 0.0
73                      0.0      0.0      0.0      0.0      0.0      180.022531
74
75 Compliance (1/GPA): -0.0026218 0.0019907 0.001617 0.0      0.0      0.0
76                      0.0019907 -0.0030583 0.0027951 0.0      0.0      0.0
77                      0.001617 0.0027951 -0.0032415 0.0      0.0      0.0
78                      0.0      0.0      0.0      0.0056518 0.0      0.0
79                      0.0      0.0      0.0      0.0      0.0055324 0.0
80                      0.0      0.0      0.0      0.0      0.0      0.0055549
81
82 Stability:
83 C11:                  63.044169998 (Stable)
84 C11C22 - C12C12:     -97263.6756949 (Unstable)
85 C11*C22*C33+2*C12*C13*C23-C11*C23*C23-C33*C12*C12: 47561899.9206 (Stable)
86 C44:                  176.934632292 (Stable)
87 C55:                  180.753497554 (Stable)
88 C66:                  180.022531471 (Stable)
89
90 Bulk Modulus B (GPA): 255.592980231
91 BR (GPA):             257.478420517
92 BV (GPA):             253.707539944
93
94 Shear Modulus G (GPA): -643.096162505
95 GR:                   -1353.7508966
96 GV:                   67.5585715904
97
98 Young's Modulus E (GPA): -11960.7418627
99
100 Poisson's Ratio v:    8.29934165372
101
102
103 L Elastic Wave V:    nan
104 T Elastic Wave V:    nan
105 M Elastic Wave V:    nan
106
107 Debye Temperature:   nan
108
109 Melting Point:       1357.0K
110
111
112
113
114
115
116
117
118 References
119 =====

```

- 120  
121 First Principles Calculations of Elastic Properties of Metals  
122 M. J. Mehl, B. M. Klein, D. A. Papaconstantopoulos  
123 1994  
124  
125 Ab Initio Study of the Elastic and Mechanical Properties of B19 TiAl  
126 Y. Wen, L. Wang, H. Liu and L. Song  
127 Crystals  
128 2017  
129  
130 Density functional theory for calculation of elastic properties of orthorhombic crystals - applications to TiSi<sub>2</sub>  
131 P. Ravindran, Lars Fast, P. A. Korzhavyi, B. Johansson  
132 Journal of Applied Physics  
133 1998
-

## M.4 BCC Iron [Ferromagnetic]

### M.4.1 Major DFT Settings

Setting	Value
ecutwfc (Ry)	71
ecutrho	430
smearing (Ry)	0.04
k-points	9 9 9 1 1 1
nspin	0 (Non-magnetic) and 2 (Collinear Spin)
pseudopotential	Fe.pbe-spn-kjpaw_psl.1.0.0.UPF
etot_conv_thr	0.0001
forc_conv_thr	0.001
conv_thr	1.0D-6
diagonalization	david
mixing_beta	0.1
mixing_mode	plain

Table M.3: Iron DFT settings

### M.4.2 Equation of State and Elastic Constants

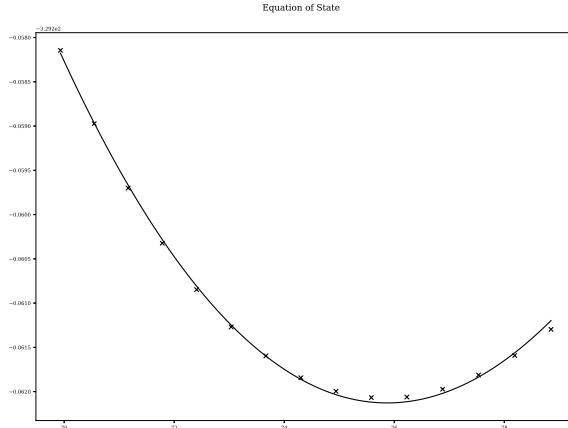


Figure M.5: Iron BCC (magnetic) equation of state plot

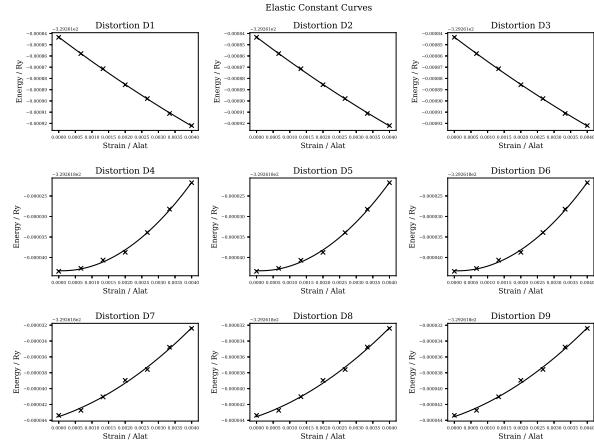


Figure M.6: Iron BCC (magnetic) elastic constants plot

### M.4.3 Iron Ferromagnetic BCC Results File

Listing M.3: Iron BCC DFT calculated properties

---

```

1 ##########
2 #          RESULTS          #
3 ##########
4
5 INPUT SETTINGS
6 #####
7
8 Structure:           bcc
9 Size:                2
10 AO:                 2.8
11 AO Units:           ang
12 AO (Bohr):          10.584
13 AO (Angs):           5.598936
14
15 Ecutfc:              71
16 Ecutrho:             430
17 Degauss:              0.04
18 Kpoints:              9 9 9 1 1 1
19 Kpoints Type:         automatic
20 EOS Points:           15
21 EOS Strain:            0.02
22 EC Points:             7
23 EC Strain:             0.004
24
25
26 RELAXED
27 #####
28
29 a0 (Bohr):           10.5935195777
30 CP:                  1.0   0.0   0.0
31                      0.0   1.0   0.0
32                      0.0   0.0   1.0
33 Volume (Bohr^3/unit cell): 1188.83291445
34 Density KG/m^3:          8430.73536197
35 AMU per crystal unit:    893.5200000000003
36 Atoms per crystal unit:  16
37 Total Energy/Ry:          -5268.18846365
38 Energy per Atom/Ry:        -329.261778978125
39 Total Force Ry/Bohr:      2.2e-05
40 Force per atom Ry/Bohr:   1.375e-06
41
42
43 a0 primitive (Bohr):     5.2967597888519995
44 a0 primitive (Ang):        2.8019859283027078
45
46
47 EOS
48 #####
49
50 V0 (Bohr^3 / atom):      75.8702665574
51 E0 (Ry / atom):           -329.262127846
52 B0 (RY/BOHR3):            0.0162627705061
53 B0 (GPA):                 239.23348553
54 BOP:                      3.31889649178
55
56
57 EC
58 #####

```

59  
 60 Stiffness (RY/BOHR3): 0.0171516 0.0124976 0.0125676 0.0 0.0 0.0  
 61 0.0124976 0.0170822 0.0125444 0.0 0.0 0.0  
 62 0.0125676 0.0125444 0.0172259 0.0 0.0 0.0  
 63 0.0 0.0 0.0 0.0095442 0.0 0.0  
 64 0.0 0.0 0.0 0.0 0.0095793 0.0  
 65 0.0 0.0 0.0 0.0 0.0 0.0095537  
 66  
 67 Stiffness (GPA): 250.132354 182.259753 183.281653 0.0 0.0 0.0  
 68 182.259753 249.120311 182.943207 0.0 0.0 0.0  
 69 183.281653 182.943207 251.216687 0.0 0.0 0.0  
 70 0.0 0.0 0.0 139.188774 0.0 0.0  
 71 0.0 0.0 0.0 0.0 139.701368 0.0  
 72 0.0 0.0 0.0 0.0 0.0 139.328420  
 73  
 74 Compliance (1/GPA): 0.0104302 -0.0043908 -0.0044121 0.0 0.0 0.0  
 75 -0.0043908 0.0104768 -0.0044261 0.0 0.0 0.0  
 76 -0.0044121 -0.0044261 0.0104228 0.0 0.0 0.0  
 77 0.0 0.0 0.0 0.0071845 0.0 0.0  
 78 0.0 0.0 0.0 0.0 0.0071581 0.0  
 79 0.0 0.0 0.0 0.0 0.0 0.0071773  
 80  
 81 Stability:  
 82 C11: 250.132354807 (Stable)  
 83 C11C22 - C12C12: 29094.4322943 (Stable)  
 84 C11\*C22\*C33+2\*C12\*C13\*C23-C11\*C23\*C23-C33\*C12\*C12: 11159910.6986 (Stable)  
 85 C44: 139.188774163 (Stable)  
 86 C55: 139.701368243 (Stable)  
 87 C66: 139.328420919 (Stable)  
 88  
 89 Bulk Modulus B (GPA): 205.266912094  
 90 BR (GPA): 205.262870468  
 91 BV (GPA): 205.270953721  
 92  
 93 Shear Modulus G (GPA): 79.4449473647  
 94 GR: 61.7805328168  
 95 GV: 97.1093619125  
 96  
 97 Young's Modulus E (GPA): 211.100586016  
 98  
 99 Poisson's Ratio v: 0.328596676183  
 100  
 101  
 102 L Elastic Wave V: 0.192124405208  
 103 T Elastic Wave V: 0.0970734382597  
 104 M Elastic Wave V: 0.108830261337  
 105  
 106 Debye Temperature: 0.000728462251696  
 107  
 108 Melting Point: 2076.0K  
 109  
 110  
 111  
 112  
 113  
 114  
 115  
 116  
 117 References  
 118 =====  
 119

- 120 First Principles Calculations of Elastic Properties of Metals  
121 M. J. Mehl, B. M. Klein, D. A. Papaconstantopoulos  
122 1994  
123  
124 Ab Initio Study of the Elastic and Mechanical Properties of B19 TiAl  
125 Y. Wen, L. Wang, H. Liu and L. Song  
126 Crystals  
127 2017  
128  
129 Density functional theory for calculation of elastic properties of orthorhombic crystals - applications to TiSi<sub>2</sub>  
130 P. Ravindran, Lars Fast, P. A. Korzhavyi, B. Johansson  
131 Journal of Applied Physics  
132 1998
-

#### M.4.4 Known and Calculated Results

	Fe Experimental	Fe DFT No Magnetism (this work)	Fe DFT Ferromagnetic (this work)
Structure	Body Centered Cubic	Body Centered Cubic	2.80 Angstrom
$a_0$ (Angs)	2.86 Angstrom[118]	2.75 Angstrom	2.42 Angstrom
Nearest Neighbour	2.48 Angstrom[118]	2.38 Angstrom	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$
Basis vectors	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$		
$E_{coh}$ (eV)	-4.32 [118]	Not Calculated	
Bulk Modulus $B_0$ (GPA)	170	287.3	239.2
Bulk Modulus $B_{0,r}$ (GPA)	-	257.5	205.3
Bulk Modulus $B_{0,g}$ (GPA)	-	253.7	205.3
Young's modulus $E$ (GPA)	211	-11960.7	211.1
Shear Modulus $G$ (GPA)	82	-643.1	79.4
Poisson Ratio $\nu$	0.29	8.29	0.33
Elastic Constants (GPA)	$\begin{bmatrix} 243 & 145 & 145 & 0 & 0 & 0 \\ 145 & 243 & 145 & 0 & 0 & 0 \\ 145 & 145 & 243 & 0 & 0 & 0 \\ 0 & 0 & 0 & 116 & 0 & 0 \\ 0 & 0 & 0 & 0 & 116 & 0 \\ 0 & 0 & 0 & 0 & 0 & 116 \end{bmatrix}$ [118]	$\begin{bmatrix} 63.0 & 329.2 & 315.3 & 0 & 0 & 0 \\ 329.2 & 176.6 & 316.4 & 0 & 0 & 0 \\ 315.3 & 316.5 & 121.7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 176.9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 180.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 180.0 \end{bmatrix}$	$\begin{bmatrix} 250.1 & 182.3 & 183.3 & 0 & 0 & 0 \\ 182.3 & 249.1 & 182.9 & 0 & 0 & 0 \\ 183.2 & 182.9 & 251.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 139.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 139.7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 139.3 \end{bmatrix}$

Table M.4: BCC Iron: Experimental Values vs DFT Calculated Variables

## M.5 FCC Iron [Antiferromagnetic]

### M.5.1 Major DFT Settings

Setting	Value
ecutwfc (Ry)	71
ecutrho	430
smearing (Ry)	0.04
k-points	9 9 9 1 1 1
nspin	2 (Collinear Spin)
pseudopotential	Fe.pbe-spn-kjpaw_psl.1.0.0.UPF
etot_conv_thr	0.0001
forc_conv_thr	0.001
conv_thr	1.0D-6
diagonalization	david
mixing_beta	0.1
mixing_mode	plain

Table M.5: Iron DFT settings

### M.5.2 Equation of State and Elastic Constants

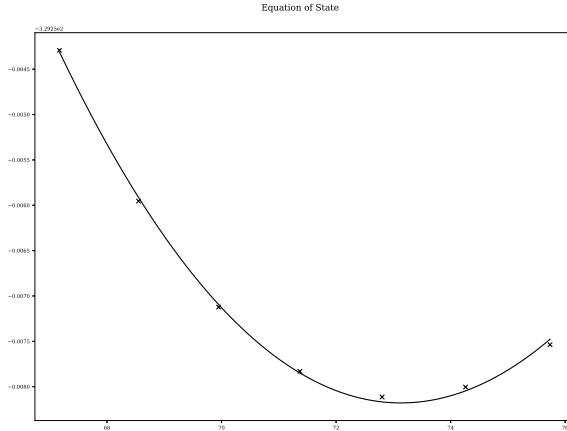


Figure M.7: Iron FCC (magnetic) equation of state plot

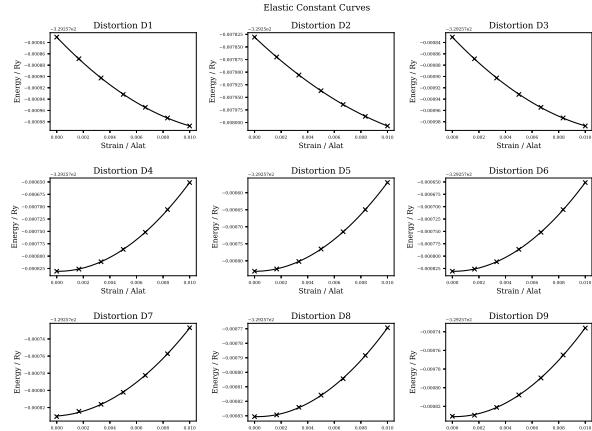


Figure M.8: Iron FCC (magnetic) elastic constants plot

### M.5.3 Iron FCC Results File

Listing M.4: Iron FCC DFT Calculated Properties

---

```

1 ######
2 #          RESULTS          #
3 ######
4
5 INPUT SETTINGS
6 #####
7
8 Structure:           fcc
9 Size:                2
10 AO:                 3.4278156
11 AO Units:           ang
12 AO (Bohr):          12.957142968
13 AO (Angs):           6.854328630072
14
15 Ecutfc:              71
16 Ecutrho:             430
17 Degauss:              0.04
18 Kpoints:              9 9 9 1 1 1
19 Kpoints Type:         automatic
20 EOS Points:           7
21 EOS Strain:            0.02
22 EC Points:             7
23 EC Strain:              0.01
24
25
26 RELAXED
27 #####
28
29 a0 (Bohr):           12.9381866998
30 CP:                  1.0    0.0    0.0
31                      0.0    1.05447369502   0.0
32                      0.0    0.0    0.999986980953
33 Volume (Bohr^3/unit cell): 2283.75934496
34 Density KG/m^3:        8777.4009231
35 AMU per crystal unit: 1787.040000000006
36 Atoms per crystal unit: 32
37 Total Energy/Ry:       -10536.25057837
38 Energy per Atom/Ry:    -329.2578305740625
39 Total Force Ry/Bohr:   5.1e-05
40 Force per atom Ry/Bohr: 1.59375e-06
41
42
43 a0 primitive (Bohr):   6.4690933498955
44 a0 primitive (Ang):     3.4221503820947197
45
46
47 EOS
48 #####
49
50 V0 (Bohr^3 / atom):   73.1321918692
51 E0 (Ry / atom):        -329.258178984
52 B0 (RY/BOHR3):          0.0153732026949
53 B0 (GPA):               226.147498244
54 BOP:                   3.74220549488
55
56
57 EC
58 #####

```

59  
60 Stiffness (RY/BOHR3): 0.0250005 0.009711 0.0160292 0.0 0.0 0.0  
61 0.009711 0.0204815 0.0089411 0.0 0.0 0.0  
62 0.0160292 0.0089411 0.0250019 0.0 0.0 0.0  
63 0.0 0.0 0.0 0.0127715 0.0 0.0  
64 0.0 0.0 0.0 0.0 0.0182966 0.0  
65 0.0 0.0 0.0 0.0 0.0 0.0127778  
66  
67 Stiffness (GPA): 364.598330 141.621966 233.764448 0.0 0.0 0.0  
68 141.621966 298.695202 130.394083 0.0 0.0 0.0  
69 233.764448 130.394083 364.619131 0.0 0.0 0.0  
70 0.0 0.0 0.0 186.254349 0.0 0.0  
71 0.0 0.0 0.0 0.0 266.830684 0.0  
72 0.0 0.0 0.0 0.0 0.0 186.346168  
73  
74 Compliance (1/GPA): 0.004966 -0.0011431 -0.002775 0.0 0.0 0.0  
75 -0.0011431 0.0042304 -0.00078 0.0 0.0 0.0  
76 -0.002775 -0.00078 0.0048006 0.0 0.0 0.0  
77 0.0 0.0 0.0 0.005369 0.0 0.0  
78 0.0 0.0 0.0 0.0 0.0037477 0.0  
79 0.0 0.0 0.0 0.0 0.0 0.0053664  
80  
81 Stability:  
82 C11: 364.598330439 (Stable)  
83 C11C22 - C12C12: 88846.9908456 (Stable)  
84 C11\*C22\*C33+2\*C12\*C13\*C23-C11\*C23\*C23-C33\*C12\*C12: 34829887.0518 (Stable)  
85 C44: 186.254349116 (Stable)  
86 C55: 266.830684551 (Stable)  
87 C66: 186.346168623 (Stable)  
88  
89 Bulk Modulus B (GPA): 221.980853006  
90 BR (GPA): 217.35352142  
91 BV (GPA): 226.608184591  
92  
93 Shear Modulus G (GPA): 144.783544731  
94 GR: 126.872037967  
95 GV: 162.695051494  
96  
97 Young's Modulus E (GPA): 356.782113599  
98  
99 Poisson's Ratio v: 0.232122456533  
100  
101  
102 L Elastic Wave V: 0.217447522685  
103 T Elastic Wave V: 0.128433002219  
104 M Elastic Wave V: 0.142291445915  
105  
106 Debye Temperature: 0.0009653163572  
107  
108 Melting Point: 2528.0K  
109  
110  
111  
112  
113  
114  
115  
116  
117 References  
118 =====  
119

120 First Principles Calculations of Elastic Properties of Metals  
 121 M. J. Mehl, B. M. Klein, D. A. Papaconstantopoulos  
 122 1994  
 123  
 124 Ab Initio Study of the Elastic and Mechanical Properties of B19 TiAl  
 125 Y. Wen, L. Wang, H. Liu and L. Song  
 126 Crystals  
 127 2017  
 128  
 129 Density functional theory for calculation of elastic properties of orthorhombic crystals - applications to TiSi<sub>2</sub>  
 130 P. Ravindran, Lars Fast, P. A. Korzhavyi, B. Johansson  
 131 Journal of Applied Physics  
 132 1998

---

Iron FCC (this work)		
Structure	Face Centered Cubic	Face Centered Cubic
$a_0$ (Angs)	-	3.42 Angstrom
Nearest Neighbour	-	2.86 Angstrom
Basis vectors	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$
$E_{coh}$ (eV)	-	-4.26
Bulk Modulus $B_0$ (GPA)	-	226.1
Bulk Modulus $B_{0,r}$ (GPA)	-	217.3
Bulk Modulus $B_{0,v}$ (GPA)	-	226.6
Young's modulus $E$ (GPA)	-	356.8
Shear Modulus $G$ (GPA)	-	144.8
Poisson Ratio $\nu$	-	0.23
Elastic Constants (GPA)	-	$\begin{bmatrix} 364.6 & 141.6 & 233.8 & 0 & 0 & 0 \\ 141.6 & 298.7 & 130.4 & 0 & 0 & 0 \\ 233.8 & 130.4 & 364.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 186.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 266.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 186.3 \end{bmatrix}$

Table M.6: Iron FCC: DFT Calculated Properties

## M.6 FCC Palladium

### M.6.1 Major DFT Settings

Setting	Value
Ecutwfc (Ry)	71
Ecutrho	430
Smearing (Ry)	0.04
K-points	9 9 9 1 1 1
Nspin	0 (Non-magnetic)
Pseudopotential	Pd.pbe-spn-kjpaw_psl.1.0.0.UPF
etot_conv_thr	0.0001
forc_conv_thr	0.001
conv_thr	1.0D-6
diagonalization	david
mixing_beta	0.1
mixing_mode	plain

Table M.7: Palladium DFT settings

### M.6.2 Equation of State and Elastic Constants

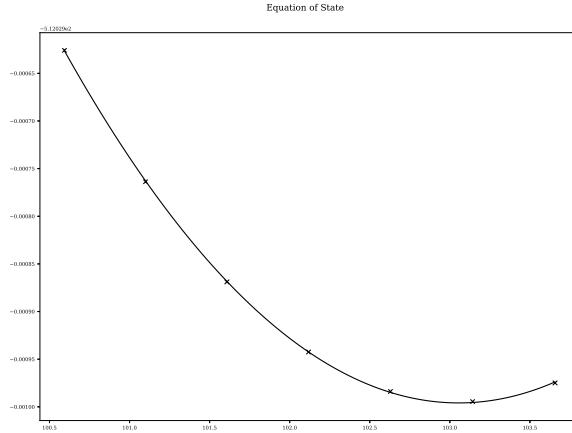


Figure M.9: Palladium FCC equation of state plot

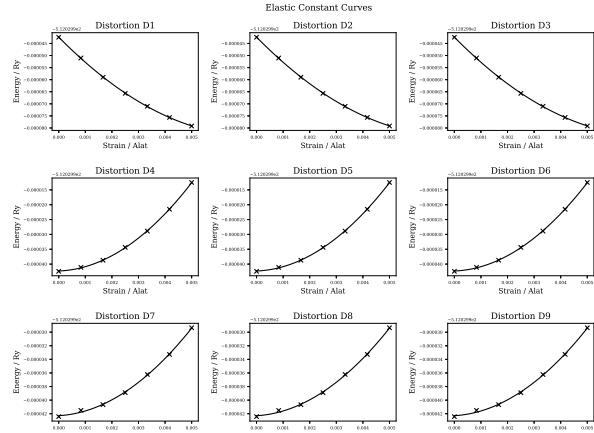


Figure M.10: Palladium FCC elastic constants plot

### M.6.3 Palladium FCC Results File

Listing M.5: Palladium FCC DFT Calculated Properties

---

```

1 ######
2 #          RESULTS          #
3 ######
4
5 INPUT SETTINGS
6 #####
7
8 Structure:          fcc
9 Size:               2
10 AO:                3.89
11 AO Units:          ang
12 AO (Bohr):         14.7042
13 AO (Angs):          7.7785218
14
15 Ecutwfc:            71
16 Ecutrho:             430
17 Degauss:            0.04
18 Kpoints:             9 9 9 1 1 1
19 Kpoints Type:        automatic
20 EOS Points:          7
21 EOS Strain:          0.005
22 EC Points:            7
23 EC Strain:           0.005
24
25
26 RELAXED
27 #####
28
29 a0 (Bohr):          14.8394198232
30 CP:                 1.0   0.0   0.0
31                  0.0   1.0   0.0
32                  0.0   0.0   1.0
33 Volume (Bohr^3/unit cell): 3267.76460963
34 Density KG/m^3:        11689.7296625
35 AMU per crystal unit:  3405.4400000000014
36 Atoms per crystal unit: 32
37 Total Energy/Ry:       -16384.95810846
38 Energy per Atom/Ry:    -512.029940889375
39 Total Force Ry/Bohr:   0.000188
40 Force per atom Ry/Bohr: 5.875e-06
41
42
43 a0 primitive (Bohr):    7.4197099116
44 a0 primitive (Ang):      3.9250265432364
45
46
47 EOS
48 #####
49
50 V0 (Bohr^3 / atom):    103.055133214
51 E0 (Ry / atom):         -512.029995887
52 B0 (RY/BOHR3):          0.012538146479
53 B0 (GPA):               184.442403779
54 BOP:                   4.7014295893
55
56
57 EC
58 #####

```

59  
60 Stiffness (RY/BOHR3): 0.0149806 0.010383 0.0103812 0.0 0.0 0.0  
61 0.010383 0.0149801 0.0103722 0.0 0.0 0.0  
62 0.0103812 0.0103722 0.0149778 0.0 0.0 0.0  
63 0.0 0.0 0.0 0.0055032 0.0 0.0  
64 0.0 0.0 0.0 0.0 0.0055036 0.0  
65 0.0 0.0 0.0 0.0 0.0 0.0055024  
66  
67 Stiffness (GPA): 218.471929 151.422406 151.395627 0.0 0.0 0.0  
68 151.422406 218.464281 151.264808 0.0 0.0 0.0  
69 151.395627 151.264808 218.430622 0.0 0.0 0.0  
70 0.0 0.0 0.0 80.2564574 0.0 0.0  
71 0.0 0.0 0.0 0.0 80.2629605 0.0  
72 0.0 0.0 0.0 0.0 0.0 80.2449828  
73  
74 Compliance (1/GPA): 0.0105867 -0.0043366 -0.0043346 0.0 0.0 0.0  
75 -0.0043366 0.0105705 -0.0043144 0.0 0.0 0.0  
76 -0.0043346 -0.0043144 0.0105702 0.0 0.0 0.0  
77 0.0 0.0 0.0 0.0124601 0.0 0.0  
78 0.0 0.0 0.0 0.0 0.012459 0.0  
79 0.0 0.0 0.0 0.0 0.0 0.0124618  
80  
81 Stability:  
82 C11: 218.471929458 (Stable)  
83 C11C22 - C12C12: 24799.5680651 (Stable)  
84 C11\*C22\*C33+2\*C12\*C13\*C23-C11\*C23\*C23-C33\*C12\*C12: 7353517.38556 (Stable)  
85 C44: 80.256457365 (Stable)  
86 C55: 80.262960524 (Stable)  
87 C66: 80.244982752 (Stable)  
88  
89 Bulk Modulus B (GPA): 173.725818131  
90 BR (GPA): 173.725800894  
91 BV (GPA): 173.725835369  
92  
93 Shear Modulus G (GPA): 56.5595512391  
94 GR: 51.5472895795  
95 GV: 61.5718128987  
96  
97 Young's Modulus E (GPA): 153.067378318  
98  
99 Poisson's Ratio v: 0.35315234089  
100  
101  
102 L Elastic Wave V: 0.145988361768  
103 T Elastic Wave V: 0.0695585857774  
104 M Elastic Wave V: 0.0782389075958  
105  
106 Debye Temperature: 0.000471028127273  
107  
108 Melting Point: 1910.0K  
109  
110  
111  
112  
113  
114  
115  
116  
117 References  
118 =====  
119

120 First Principles Calculations of Elastic Properties of Metals  
 121 M. J. Mehl, B. M. Klein, D. A. Papaconstantopoulos  
 122 1994  
 123  
 124 Ab Initio Study of the Elastic and Mechanical Properties of B19 TiAl  
 125 Y. Wen, L. Wang, H. Liu and L. Song  
 126 Crystals  
 127 2017  
 128  
 129 Density functional theory for calculation of elastic properties of orthorhombic crystals - applications to TiSi<sub>2</sub>  
 130 P. Ravindran, Lars Fast, P. A. Korzhavyi, B. Johansson  
 131 Journal of Applied Physics  
 132 1998

---

#### M.6.4 Known and Calculated Values

	Palladium FCC	Palladium FCC (this work)
Structure	Face Centered Cubic	Face Centered Cubic
$a_0$ (Angs)	3.89	3.92
Nearest Neighbour	2.75	2.77
Basis vectors	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$
$E_{coh}$ (eV)	-3.91	not calculated
Bulk Modulus $B_0$ (GPA)	195.5	184.4
Bulk Modulus $B_{0,r}$ (GPA)	-	173.7
Bulk Modulus $B_{0,v}$ (GPA)	-	173.7
Young's modulus $E$ (GPA)	121	153.1
Shear Modulus $G$ (GPA)	44	56.6
Poisson Ratio $\nu$	0.39	0.35
Elastic Constants (GPA)	$\begin{bmatrix} 234 & 176.1 & 176.1 & 0 & 0 & 0 \\ 176.1 & 234 & 176.1 & 0 & 0 & 0 \\ 176.1 & 176.1 & 234 & 0 & 0 & 0 \\ 0 & 0 & 0 & 71.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 71.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 71.2 \end{bmatrix}$	$\begin{bmatrix} 218.5 & 151.4 & 151.4 & 0 & 0 & 0 \\ 151.4 & 218.5 & 151.3 & 0 & 0 & 0 \\ 151.4 & 151.3 & 218.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 80.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 80.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80.3 \end{bmatrix}$

Table M.8: FCC Palladium: Experimental Values vs DFT Calculated Properties

## M.7 FCC Ruthenium

### M.7.1 Major DFT Settings

Ecutwfc: 71  
 Ecutrho: 430  
 Smearing: 0.04  
 K-points: 9 9 9 1 1 1  
 Nspin: 1 (non-polarized calculation)  
 Pseudopotential: Ru 101.07 Ru.pbe-spn-kjpaw\_psl.1.0.0.UPF

### M.7.2 Equation of State and Elastic Constants

Setting	Value
ecutwfc (Ry)	71
ecutrho	430
smearing (Ry)	0.04
k-points	9 9 9 1 1 1
nspin	2 (Collinear Spin)
pseudopotential	Ru.pbe-spn-kjpaw_psl.1.0.0.UPF
etot_conv_thr	0.0001
forc_conv_thr	0.001
conv_thr	1.0D-6
diagonalization	david
mixing_beta	0.1
mixing_mode	plain

Table M.9: Ruthenium DFT settings

### M.7.3 Ruthenium FCC Results File

Listing M.6: Ruthenium FCC DFT Calculated Properties

---

```

1 ######
2 #          RESULTS          #
3 #####
4 Fri, 26 Nov 2021 04:44:37 +0000
5
6 INPUT SETTINGS
7 #####
8
9 Structure Type:      fcc
10 Size:                2
11 AO:                  4.0
12 AO Units:            ang
13 AO (Bohr):           15.12
14 AO (Angs):            7.99848
15
16 Ecutwfc:              71
17 Ecutrho:               430
18 Degauss:                0.04
19 Kpoints:                9 9 9 1 1 1
20 Kpoints Type:          automatic
21 EOS Points:             11
22 EOS Strain:              0.005
23 EC Points:                9
24 EC Strain:                0.005
25
26
27 RELAXED
28 #####
29
30 a0 (Bohr):            14.4007416
31 CP:                   1.0    0.0    0.0
32                      0.0    1.0    0.0
33                      0.0    0.0    1.0
34 Volume (Bohr^3/unit cell): 2986.445358287136
35 Density KG/m^3:          12147.856996566989
36 AMU per crystal unit:   3234.240000000001
37 Atoms per crystal unit:  32
38 Total Energy/Ry:         -13813.30088106
39 Energy per Atom/Ry:      -431.665652533125
40 Total Force Ry/Bohr:     0.000137
41 Force per atom Ry/Bohr:  4.28125e-06
42
43
44 a0 primitive (Bohr):     7.2003708
45 a0 primitive (Ang):       3.8089961532000003
46
47
48 EOS
49 #####
50
51 V0 (Bohr^3 / atom):      93.41718637560044
52 E0 (Ry / atom):           -431.66565367812325
53 B0 (RY/BOHR3):            0.020926676273491183
54 B0 (GPA):                 307.84187132119206
55 BOP:                      2.2239400402684173
56
57
58 EC

```

59 #####  
60  
61 Stiffness (RY/BOHR3): 0.0323332 0.0150236 0.0150295 0.0 0.0 0.0  
62 0.0150236 0.032334 0.0150304 0.0 0.0 0.0  
63 0.0150295 0.0150304 0.0323458 0.0 0.0 0.0  
64 0.0 0.0 0.0 0.0167987 0.0 0.0  
65 0.0 0.0 0.0 0.0 0.0167978 0.0  
66 0.0 0.0 0.0 0.0 0.0 0.0167975  
67  
68 Stiffness (GPA): 471.535040 219.099143 219.185184 0.0 0.0 0.0  
69 219.099143 471.546672 219.197356 0.0 0.0 0.0  
70 219.185184 219.197356 471.719577 0.0 0.0 0.0  
71 0.0 0.0 0.0 244.986869 0.0 0.0  
72 0.0 0.0 0.0 0.0 244.973541 0.0  
73 0.0 0.0 0.0 0.0 0.0 244.969279  
74  
75 Compliance (1/GPA): 0.0030075 -0.0009538 -0.0009542 0.0 0.0 0.0  
76 -0.0009538 0.0030075 -0.0009543 0.0 0.0 0.0  
77 -0.0009542 -0.0009543 0.0030067 0.0 0.0 0.0  
78 0.0 0.0 0.0 0.0040819 0.0 0.0  
79 0.0 0.0 0.0 0.0 0.0040821 0.0  
80 0.0 0.0 0.0 0.0 0.0 0.0040821  
81  
82 Stability:  
83 C11: 471.535040254 (Stable)  
84 C11C22 - C12C12: 174346.3443625905 (Stable)  
85 C11\*C22\*C33+2\*C12\*C13\*C23-C11\*C23\*C23-C33\*C12\*C12: 80639667.71993023 (Stable)  
86 C44: 244.986869322 (Stable)  
87 C55: 244.97354179 (Stable)  
88 C66: 244.969279805 (Stable)  
89  
90 Bulk Modulus B (GPA): 303.30717357998026  
91 BR (GPA): 303.3071627547384  
92 BV (GPA): 303.3071844052222  
93  
94 Shear Modulus G (GPA): 187.73207685789976  
95 GR: 177.99024187226613  
96 GV: 197.47391184353336  
97  
98 Young's Modulus E (GPA): 466.87258322673495  
99  
100 Poisson's Ratio v: 0.24345447789438046  
101  
102  
103 L Elastic Wave V: 0.21347878392203118  
104 T Elastic Wave V: 0.12431381857023048  
105 M Elastic Wave V: 0.1379068118794884  
106  
107 Debye Temperature: 0.0008555430077179703  
108  
109 Melting Point: 3072.0K  
110  
111  
112  
113  
114  
115  
116  
117  
118 References  
119 =====

120  
 121 First Principles Calculations of Elastic Properties of Metals  
 122 M. J. Mehl, B. M. Klein, D. A. Papaconstantopoulos  
 123 1994  
 124  
 125 Ab Initio Study of the Elastic and Mechanical Properties of B19 TiAl  
 126 Y. Wen, L. Wang, H. Liu and L. Song  
 127 Crystals  
 128 2017  
 129  
 130 Density functional theory for calculation of elastic properties of orthorhombic crystals - applications to TiSi<sub>2</sub>  
 131 P. Ravindran, Lars Fast, P. A. Korzhavyi, B. Johansson  
 132 Journal of Applied Physics  
 133 1998

---

#### M.7.4 Calculated Results

	Ruthenium FCC (this work)						
Structure	Face Centered Cubic						Face Centered Cubic
$a_0$ (Angs)	-						3.81 Angstrom
Nearest Neighbour	-						2.69 Angstrom
Basis vectors	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$						$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$
$E_{coh}$ (eV)	-						-6.62
Bulk Modulus $B_0$ (GPA)	-						307.8
Bulk Modulus $B_{0,r}$ (GPA)	-						303.3
Bulk Modulus $B_{0,v}$ (GPA)	-						303.3
Young's modulus $E$ (GPA)	-						466.9
Shear Modulus $G$ (GPA)	-						187.7
Poisson Ratio $\nu$	-						0.24
Elastic Constants (GPA)	$\begin{bmatrix} 471.54 & 219.10 & 219.20 & 0 & 0 & 0 \\ 219.10 & 471.55 & 219.20 & 0 & 0 & 0 \\ 219.19 & 219.20 & 471.72 & 0 & 0 & 0 \\ 0 & 0 & 0 & 244.99 & 0 & 0 \\ 0 & 0 & 0 & 0 & 244.97 & 0 \\ 0 & 0 & 0 & 0 & 0 & 244.97 \end{bmatrix}$						

Table M.10: Ruthenium FCC: DFT Calculated Properties

## Appendix N

# Transferability of Interatomic Potentials

### N.1 Sheng Aluminium Potential

#### N.1.1 FCC Structure

The EAMPA code was used to compute both the Birch-Murnaghan and Rose-Vinet equations of state using the known values for Aluminium and the Sheng Al potential[33].

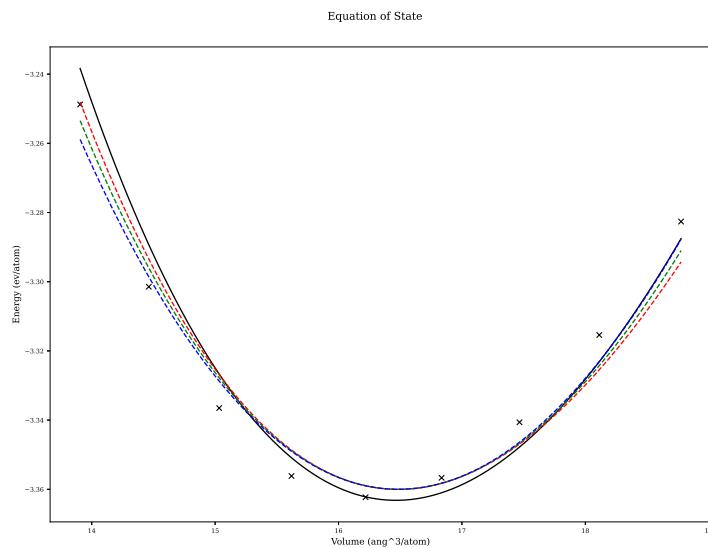


Figure N.1: Birch Murnaghan equation of state for FCC Aluminium - comparison of experimental data to the Sheng Al potential

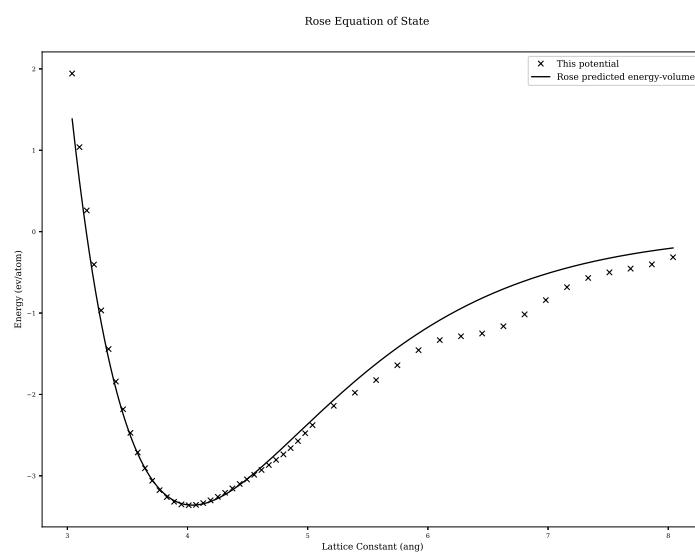


Figure N.2: Rose-Vinet equation of state for FCC Aluminium - comparison of experimental data to the Sheng Al potential

### N.1.2 BCC Structure

## N.2 Mendelev et al 2003 Iron Potential

The EAMPA code was used to compute both the Birch-Murnaghan and Rose-Vinet equations of state using the known values for Aluminium and the Sheng Al potential[118].

### N.2.1 BCC Structure

### N.2.2 FCC Structure

## N.3 Ackland et al 1997 Iron Potential

The EAMPA code was used to compute both the Birch-Murnaghan and Rose-Vinet equations of state using the known values for Aluminium and the Sheng Al potential[Fe·abch].

### N.3.1 BCC Structure

### N.3.2 FCC Structure

The EAMPA code was used to compute both the Birch-Murnaghan and Rose-Vinet equations of state using the known values for Aluminium and the Sheng Al potential[33].

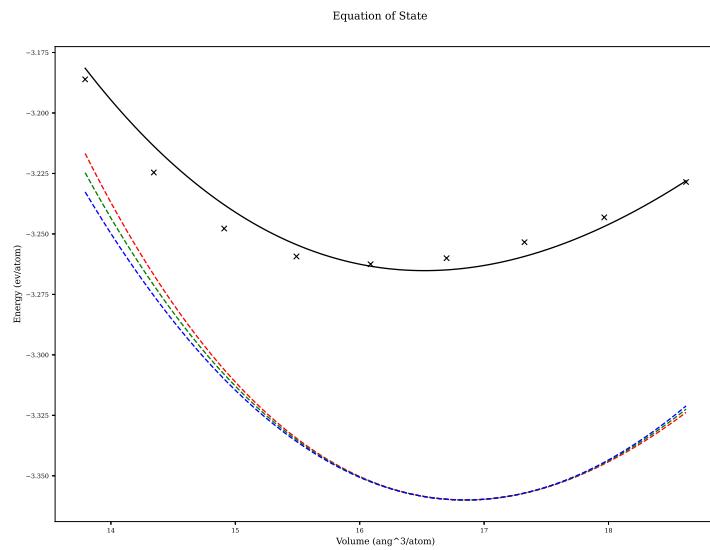


Figure N.3: Birch Murnaghan equation of state for BCC Aluminium - comparison of experimental data to the Sheng Al potential

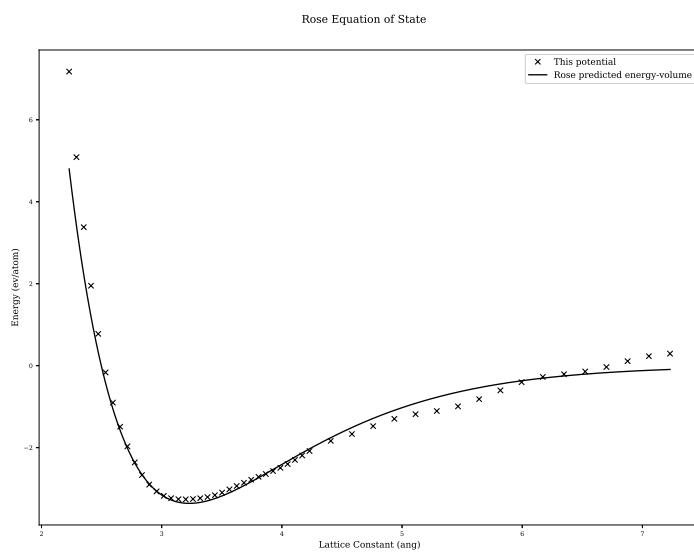


Figure N.4: Rose-Vinet equation of state for BCC Aluminium - comparison of experimental data to the Sheng Al potential

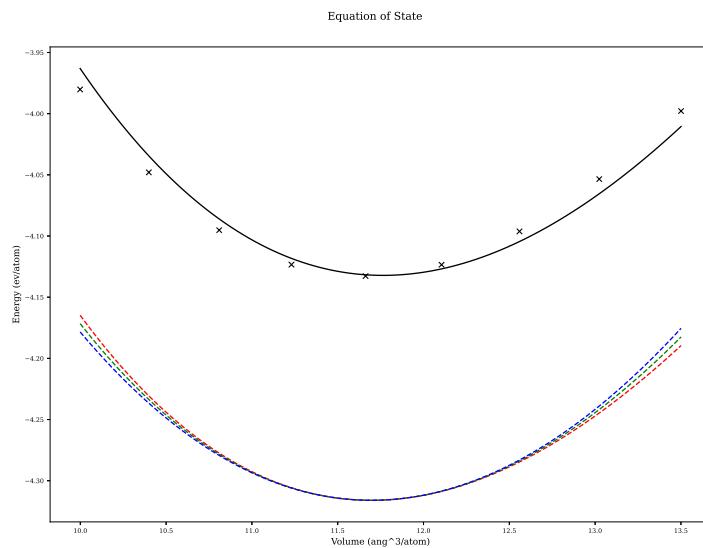


Figure N.5: Birch Murnaghan equation of state for FCC Aluminium - comparison of experimental data to the Sheng Al potential

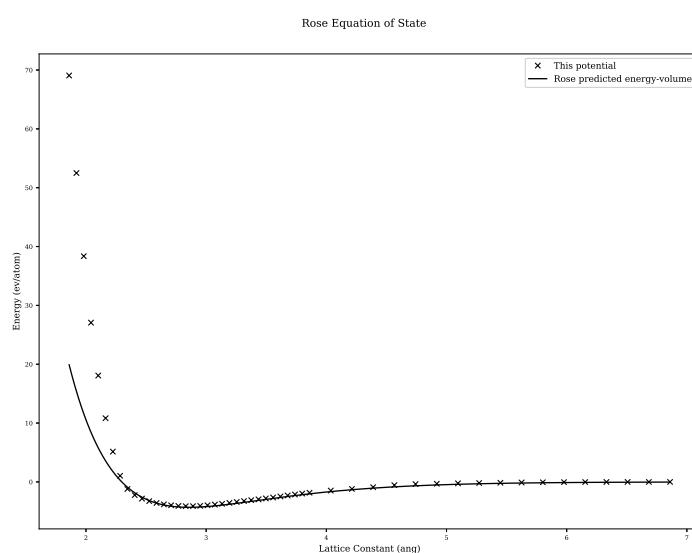


Figure N.6: Rose-Vinet equation of state for FCC Aluminium - comparison of experimental data to the Sheng Al potential

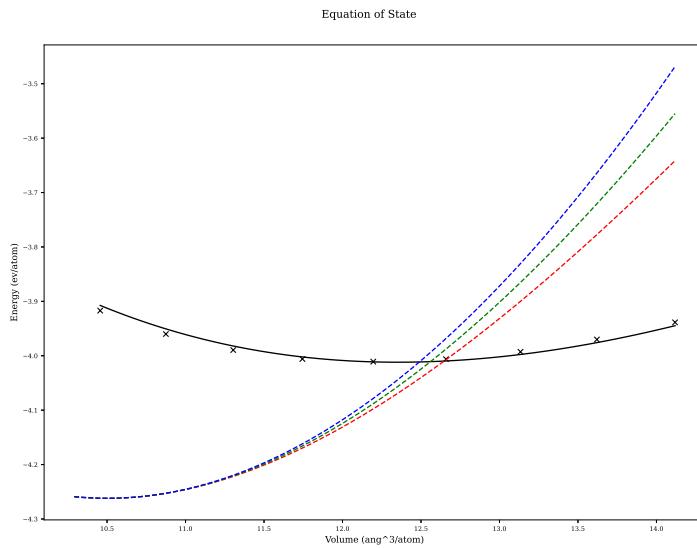


Figure N.7: Birch Murnaghan equation of state for FCC Aluminium - comparison of experimental data to the Sheng Al potential

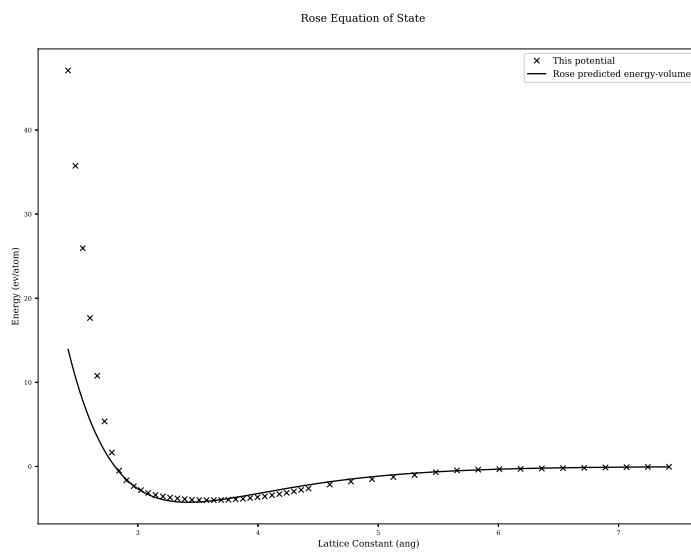


Figure N.8: Rose-Vinet equation of state for FCC Aluminium - comparison of experimental data to the Sheng Al potential

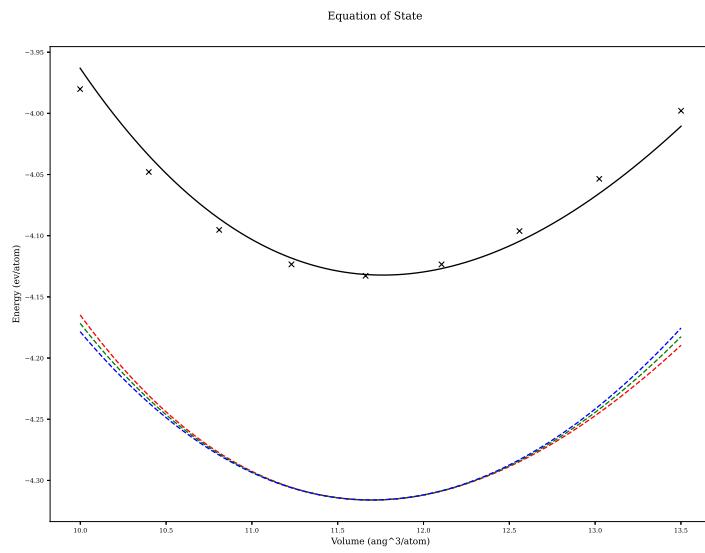


Figure N.9: Birch Murnaghan equation of state for FCC Aluminium - comparison of experimental data to the Sheng Al potential

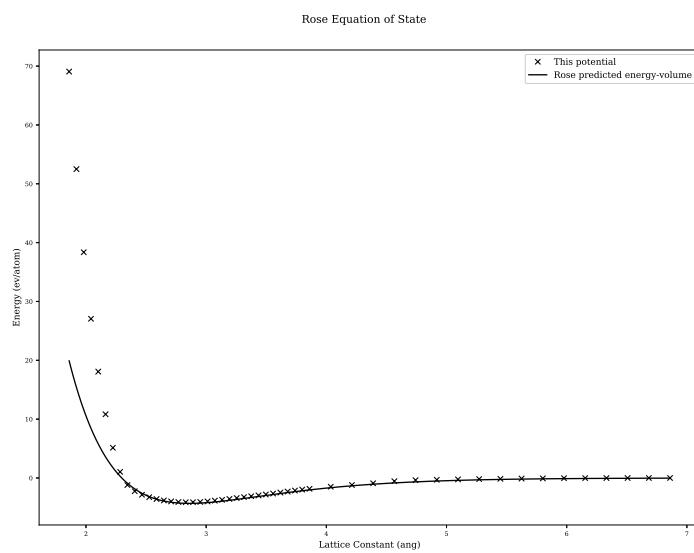


Figure N.10: Rose-Vinet equation of state for FCC Aluminium - comparison of experimental data to the Sheng Al potential

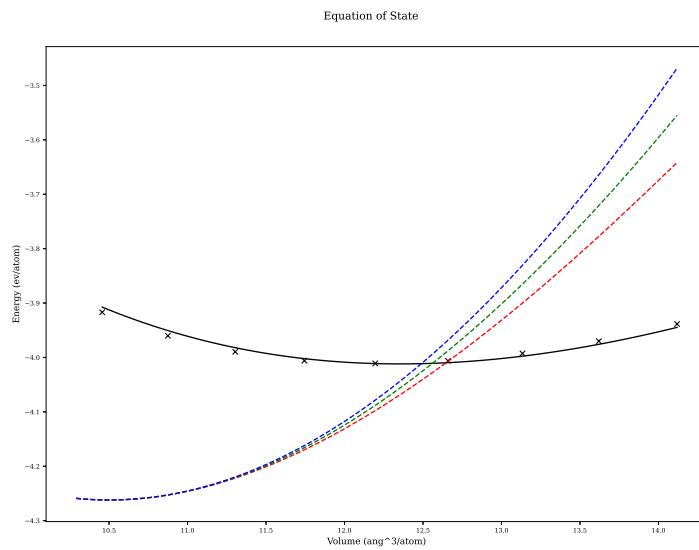


Figure N.11: Birch Murnaghan equation of state for FCC Aluminium - comparison of experimental data to the Sheng Al potential

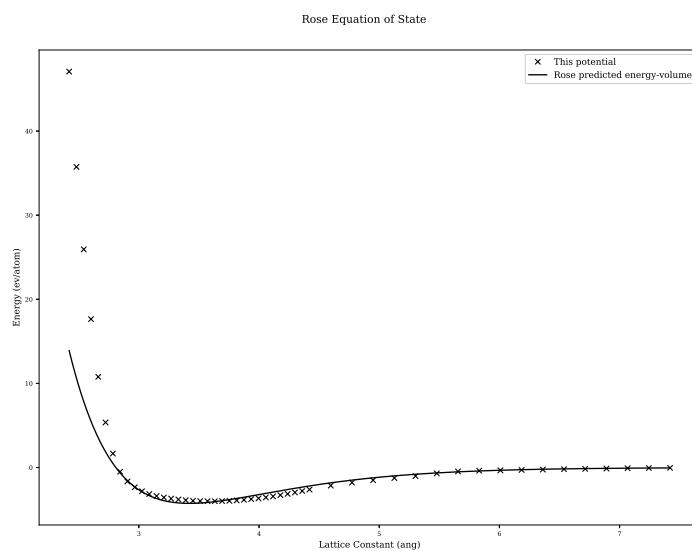


Figure N.12: Rose-Vinet equation of state for FCC Aluminium - comparison of experimental data to the Sheng Al potential

# Appendix O

## EAMPA Code

The full code is available at [https://github.com/BenPalmer1983/eampa\\_v3](https://github.com/BenPalmer1983/eampa_v3)

Several of the important functions and subroutines that make up the code are listed below. In some cases they will require Fortran module globals or Python classes in order to run.

### O.1 Energy, Force, and Stress Subroutine

The subroutine that calculates energy, stress and forces is given in listing ???. It uses OpenMP to split the calculations for individual configurations so each is computed by an individual process, meaning multiple configurations may be calculated in parallel. It handles one or multiple density/embedding groups where either standard EAM, 2BEAM or more bands are required.

Listing O.1: Energy, Force and Stress subroutine (Fortran)

```
1 SUBROUTINE energy_force_stress()
2 !#####
3 INTEGER(KIND=StandardInteger) :: cn
4 !#####
5
6 CALL start_t()
7 !$OMP PARALLEL
8 !$OMP DO
9 DO cn = 1, cc
10   CALL energy_inner(cn)
11   CALL energy_force_stress_inner(cn)
12 END DO
13 !$OMP END DO
14 !$OMP END PARALLEL
15 CALL end_t(efs_timer)
16 efs_timer_sum = efs_timer_sum + efs_timer
17 END SUBROUTINE energy_force_stress
18
19
20
21 SUBROUTINE energy_force_stress_inner(cn)
22 !#####
23 INTEGER(KIND=StandardInteger) :: cn
24 !#####
25 CALL energy_force_stress_calc(cn, &
26     nlist_l(key(cn, 5):key(cn, 6),:), &
27     nlist_r(key(cn, 5):key(cn, 6),:), &
28     labels(key(cn, 1):key(cn, 2)), &
```

```

29         coords(key(cn, 1):key(cn, 2),:) &
30     )
31 END SUBROUTINE energy_force_stress_inner
32
33
34 SUBROUTINE energy_force_stress_calc(cn, nl_l, nl_r, c_l, c_r)
35 !#####
36 INTEGER(KIND=StandardInteger) :: cn
37 INTEGER(KIND=StandardInteger) :: nl_l(:, :)
38 REAL(kind=DoubleReal) :: nl_r(:, :)
39 INTEGER(KIND=StandardInteger) :: c_l(:)
40 REAL(kind=DoubleReal) :: c_r(:, :)
41 !#####
42 INTEGER(KIND=StandardInteger) :: i, j, n, f, fn, dn
43 INTEGER(KIND=StandardInteger) :: cc, nc
44 REAL(kind=DoubleReal) :: fx, dfwdx
45 REAL(kind=DoubleReal) :: fx_arr(1:2)
46 REAL(kind=DoubleReal) :: fv(1:3)
47 REAL(kind=DoubleReal), ALLOCATABLE :: density(:, :)
48 REAL(kind=DoubleReal), ALLOCATABLE :: density_grad_ab(:, :)
49 REAL(kind=DoubleReal), ALLOCATABLE :: density_grad_ba(:, :)
50 REAL(kind=DoubleReal), ALLOCATABLE :: nl_force(:, :)
51 !REAL(kind=DoubleReal) :: s(1:3, 1:3)
52 REAL(kind=DoubleReal) :: epA, epB, dpAB, dpBA
53 !#####
54 ! nlist_l(nc, 1) = ids(an)
55 ! nlist_l(nc, 2) = labels(an)
56 ! nlist_l(nc, 3) = ghostids(gn)
57 ! nlist_l(nc, 4) = ghostlabels(gn)
58 ! nlist_r(nc, 1) = r_mag
59 ! nlist_r(nc, 2:4) = r(1:3)/r_mag
60
61 cc = SIZE(c_l, 1)
62 nc = SIZE(nl_l, 1)
63 ALLOCATE(density(1:cc, 1:fgroup_max))
64 ALLOCATE(density_grad_ab(1:nc, 1:fgroup_max))
65 ALLOCATE(density_grad_ba(1:nc, 1:fgroup_max))
66 ALLOCATE(nl_force(1:nc, 1:3))
67 config_energy(cn, :) = 0.0D0
68 config_forces(cn, 1:key(cn, 20), 1:3) = 0.0D0
69 density = 0.0D0
70 density_grad_ab = 0.0D0
71 density_grad_ba = 0.0D0
72
73 DO n = 1, nc
74   ! PAIR ENERGY
75   CALL search_f(1, nl_l(n, 2), nl_l(n, 4), 0, nl_r(n, 1), fx)
76   config_energy(cn, 1) = config_energy(cn, 1) + fx
77   ! PAIR FORCE
78   CALL search_grad(1, nl_l(n, 2), nl_l(n, 4), 0, nl_r(n, 1), dfwdx)
79   fv(:) = dfwdx * nl_r(n, 2:4)
80   config_forces(cn, nl_l(n, 1), :) = config_forces(cn, nl_l(n, 1), :) - fv(:)
81   config_forces(cn, nl_l(n, 3), :) = config_forces(cn, nl_l(n, 3), :) + fv(:)
82   nl_force(n,:) = - fv(:)
83
84   ! LOOP THROUGH DENSITY GROUPS
85   DO fn = 1, fgroup_max
86     ! DENSITY OF ATOM B AT A (B with any atom)
87     CALL search_f(2, nl_l(n, 2), 0, fn, nl_r(n, 1), fx)
88     CALL search_grad(2, nl_l(n, 2), 0, fn, nl_r(n, 1), dfwdx)
89     density(nl_l(n, 1), fn) = density(nl_l(n, 1), fn) + fx

```

```

90      density_grad_ab(n, fn) = density_grad_ab(n, fn) + dfwdx
91      ! DENSITY OF ATOM B AT A (B with only atom A)
92      CALL search_f(2, nl_l(n, 2), nl_l(n, 4), fn, nl_r(n, 1), fx)
93      CALL search_grad(2, nl_l(n, 2), nl_l(n, 4), fn, nl_r(n, 1), dfwdx)
94      density(nl_l(n, 1), fn) = density(nl_l(n, 1), fn) + fx
95      density_grad_ab(n, fn) = density_grad_ab(n, fn) + dfwdx
96      ! DENSITY OF ATOM A AT B (A with any atom)
97      CALL search_f(2, nl_l(n, 4), 0, fn, nl_r(n, 1), fx)
98      CALL search_grad(2, nl_l(n, 4), 0, fn, nl_r(n, 1), dfwdx)
99      density(nl_l(n, 3), fn) = density(nl_l(n, 3), fn) + fx
100     density_grad_ba(n, fn) = density_grad_ba(n, fn) + dfwdx
101     ! DENSITY OF ATOM A AT B (A with only atom B)
102     CALL search_f(2, nl_l(n, 4), nl_l(n, 2), fn, nl_r(n, 1), fx)
103     CALL search_grad(2, nl_l(n, 4), nl_l(n, 2), fn, nl_r(n, 1), dfwdx)
104     density(nl_l(n, 3), fn) = density(nl_l(n, 3), fn) + fx
105     density_grad_ba(n, fn) = density_grad_ba(n, fn) + dfwdx
106   END DO
107 END DO
108
109 ! EMBED ENERGY
110 DO n = 1, cc
111   ! LOOP THROUGH DENSITY GROUPS
112   DO fn = 1, fgroup_max
113     CALL search_f(3, nl_l(n, 4), 0, fn, density(n, fn), fx)
114     config_energy(cn, 2) = config_energy(cn, 2) + fx
115   END DO
116 END DO
117
118 DO n = 1, nc
119   DO fn = 1, fgroup_max
120     ! @Fi(p)/@p Gradient of embedding energy for atom A
121     CALL search_grad(3, nl_l(n, 2), 0, fn, density(nl_l(n, 1), fn), dfwdx)
122     epA = dfwdx
123
124     ! @Fi(p)/@p Gradient of embedding energy for atom A
125     CALL search_grad(3, nl_l(n, 4), 0, fn, density(nl_l(n, 3), fn), dfwdx)
126     epB = dfwdx
127
128     ! @Pij(r)/@r Gradient of density function at A due to atom B
129     dpAB = density_grad_ab(n,fn)
130
131     ! @Pji(r)/@r Gradient of density function at B due to atom A
132     dpBA = density_grad_ba(n,fn)
133
134     fv(:) = (epA * dpBA + epB * dpAB) * nl_r(n, 2:4)
135     config_forces(cn, nl_l(n, 1), :) = config_forces(cn, nl_l(n, 1), :) - fv(:)
136     config_forces(cn, nl_l(n, 3), :) = config_forces(cn, nl_l(n, 3), :) + fv(:)
137     nl_force(n,:)= nl_force(n,:)- fv(:)
138   END DO
139 END DO
140
141 ! STRESS
142 config_stresses(cn, 1:3, 1:3) = 0.0D0
143 DO n = 1, nc
144   IF(nlisthalo(n))THEN
145     DO i = 1,3
146       DO j = 1,3
147         config_stresses(cn, i,j) = config_stresses(cn, i,j) + (nl_r(n, 1+i) * nl_force(n,j))
148       END DO
149     END DO
150   END IF

```

```

151 END DO
152 config_stresses(cn, 1:3, 1:3) = config_stresses(cn, 1:3, 1:3) / (2.0D0 * volume(cn))
153 stresses_calculated(cn) = 1
154
155 max_density = MAX(max_density, MAXVAL(density))
156
157 DEALLOCATE(density)
158 DEALLOCATE(density_grad_ab)
159 DEALLOCATE(density_grad_ba)
160 DEALLOCATE(nl_force)
161 config_energy(cn, 3) = config_energy(cn, 1) + config_energy(cn, 2)
162 config_energy(cn, 4:6) = config_energy(cn, 1:3) / cc
163
164 END SUBROUTINE energy_force_stress_calc

```

---

## O.2 Bulk Properties: Add Configuration Subroutine

In this subroutine, the structures used to compute bulk properties are created. The atom positions are set, depending on the crystal type, and then distortions are applied to the basis vector for each configuration in order to compute the equation of state and elastic constants (listing ??).

Listing O.2: Bulk Properties - add configuration subroutine (Fortran)

```

1 SUBROUTINE add_bp_config(rcut_in, alat_in, uv_in, label_in, crystal_type_in, expansion_in, bp_id)
2 !#####
3 REAL(kind=DoubleReal), INTENT(IN) :: rcut_in
4 REAL(kind=DoubleReal), INTENT(IN) :: alat_in
5 REAL(kind=DoubleReal), INTENT(IN) :: uv_in(1:3,1:3)
6 INTEGER(KIND=StandardInteger), INTENT(IN) :: label_in
7 INTEGER(KIND=StandardInteger), INTENT(IN) :: crystal_type_in
8 INTEGER(KIND=StandardInteger), INTENT(IN) :: expansion_in
9 INTEGER(KIND=StandardInteger), INTENT(OUT) :: bp_id
10 !#####
11 INTEGER(KIND=StandardInteger) :: primitive_labels(1:20)
12 REAL(kind=DoubleReal) :: primitive_coords(1:20,1:3)
13 INTEGER(KIND=StandardInteger) :: atoms_per_crystal
14 !#####
15
16 ! SC = 1, BCC = 2, FCC = 3, ZB = 4
17 !#####
18
19 ! SC
20 IF(crystal_type_in .EQ. 1)THEN
21   atoms_per_crystal = 1
22   primitive_labels(1) = label_in
23   primitive_coords(1,1) = 0.0D0
24   primitive_coords(1,2) = 0.0D0
25   primitive_coords(1,3) = 0.0D0
26   CALL add_inner(rcut_in, alat_in, uv_in, label_in, crystal_type_in, expansion_in, atoms_per_crystal, &
27                 primitive_labels(1:1), primitive_coords(1:1,1:3), bp_id)
28 ! BCC
29 ELSE IF(crystal_type_in .EQ. 2)THEN
30
31   atoms_per_crystal = 2
32
33   primitive_labels(1) = label_in
34   primitive_labels(2) = label_in
35
36   primitive_coords(1,1) = 0.0D0

```

```

37 primitive_coords(1,2) = 0.0D0
38 primitive_coords(1,3) = 0.0D0
39 primitive_coords(2,1) = 0.5D0
40 primitive_coords(2,2) = 0.5D0
41 primitive_coords(2,3) = 0.5D0
42
43 CALL add_inner(rcut_in, alat_in, uv_in, label_in, crystal_type_in, expansion_in, atoms_per_crystal, &
44 primitive_labels(1:2), primitive_coords(1:2,1:3), bp_id)
45
46
47
48 ! FCC
49 ELSE IF(crystal_type_in .EQ. 3)THEN
50
51 atoms_per_crystal = 4
52
53 primitive_labels(1) = label_in
54 primitive_labels(2) = label_in
55 primitive_labels(3) = label_in
56 primitive_labels(4) = label_in
57
58 primitive_coords(1,1) = 0.0D0
59 primitive_coords(1,2) = 0.0D0
60 primitive_coords(1,3) = 0.0D0
61 primitive_coords(2,1) = 0.5D0
62 primitive_coords(2,2) = 0.5D0
63 primitive_coords(2,3) = 0.0D0
64 primitive_coords(3,1) = 0.5D0
65 primitive_coords(3,2) = 0.0D0
66 primitive_coords(3,3) = 0.5D0
67 primitive_coords(4,1) = 0.0D0
68 primitive_coords(4,2) = 0.5D0
69 primitive_coords(4,3) = 0.5D0
70
71 CALL add_inner(rcut_in, alat_in, uv_in, label_in, crystal_type_in, expansion_in, atoms_per_crystal, &
72 primitive_labels(1:4), primitive_coords(1:4,1:3), bp_id)
73
74 ! ZB
75 ELSE IF(crystal_type_in .EQ. 4)THEN
76
77 atoms_per_crystal = 8
78
79 primitive_labels(1) = label_in
80 primitive_labels(2) = label_in
81 primitive_labels(3) = label_in
82 primitive_labels(4) = label_in
83 primitive_labels(5) = label_in
84 primitive_labels(6) = label_in
85 primitive_labels(7) = label_in
86 primitive_labels(8) = label_in
87
88 primitive_coords(1,1) = 0.0D0
89 primitive_coords(1,2) = 0.0D0
90 primitive_coords(1,3) = 0.0D0
91 primitive_coords(2,1) = 0.5D0
92 primitive_coords(2,2) = 0.5D0
93 primitive_coords(2,3) = 0.0D0
94 primitive_coords(3,1) = 0.5D0
95 primitive_coords(3,2) = 0.0D0
96 primitive_coords(3,3) = 0.5D0
97 primitive_coords(4,1) = 0.0D0

```

```

98     primitive_coords(4,2) = 0.5D0
99     primitive_coords(4,3) = 0.5D0
100    primitive_coords(5,1) = 0.25D0
101    primitive_coords(5,2) = 0.25D0
102    primitive_coords(5,3) = 0.25D0
103    primitive_coords(6,1) = 0.75D0
104    primitive_coords(6,2) = 0.75D0
105    primitive_coords(6,3) = 0.25D0
106    primitive_coords(7,1) = 0.25D0
107    primitive_coords(7,2) = 0.75D0
108    primitive_coords(7,3) = 0.75D0
109    primitive_coords(8,1) = 0.75D0
110    primitive_coords(8,2) = 0.25D0
111    primitive_coords(8,3) = 0.75D0
112
113    CALL add_inner(rcut_in, alat_in, uv_in, label_in, crystal_type_in, expansion_in, atoms_per_crystal, &
114                  primitive_labels(1:8), primitive_coords(1:8,1:3), bp_id)
115
116
117 END IF
118
119 END SUBROUTINE add_bp_config
120
121
122
123
124
125 SUBROUTINE add_inner(rcut_in, alat_in, uv_in, label, crystal_type, expansion, atoms_per_crystal, &
126                      primitive_labels, primitive_coords, bp_id)
127 !#####
128 REAL(kind=DoubleReal), INTENT(IN) :: rcut_in
129 REAL(kind=DoubleReal), INTENT(IN) :: alat_in
130 REAL(kind=DoubleReal), INTENT(IN) :: uv_in(1:3,1:3)
131 INTEGER(KIND=StandardInteger), INTENT(IN) :: label
132 INTEGER(KIND=StandardInteger), INTENT(IN) :: crystal_type
133 INTEGER(KIND=StandardInteger), INTENT(IN) :: expansion
134 INTEGER(KIND=StandardInteger), INTENT(IN) :: atoms_per_crystal
135 INTEGER(KIND=StandardInteger), INTENT(IN) :: primitive_labels(:)
136 REAL(kind=DoubleReal), INTENT(IN) :: primitive_coords(:, :)
137 INTEGER(KIND=StandardInteger), INTENT(OUT) :: bp_id
138 !#####
139 INTEGER(KIND=StandardInteger) :: expanded_labels(1:1024)
140 REAL(kind=DoubleReal) :: expanded_coords(1:1024,1:3)
141 REAL(kind=DoubleReal) :: strain(1:3,1:3)
142 REAL(kind=DoubleReal) :: alat_temp
143 REAL(kind=DoubleReal) :: uv_temp(1:3,1:3)
144 REAL(kind=DoubleReal) :: uv_config(1:3,1:3)
145 REAL(kind=DoubleReal) :: vol_cell(1:3,1:3)
146 REAL(kind=DoubleReal) :: vol
147 REAL(kind=DoubleReal) :: sigma
148 INTEGER(KIND=StandardInteger) :: n, i, j, k, m, cn_counter, cc_start, cc_end, ec_n, rn, cc_counter
149 !#####
150
151 ! CHECK
152 n = 1
153 bp_id = 0
154 DO WHILE(bp_keys_i(n, 1) .NE. -1)
155   IF(bp_keys_i(n, 1) .EQ. label .AND. &
156      bp_keys_i(n, 2) .EQ. crystal_type .AND. &
157      bp_keys_i(n, 3) .EQ. expansion .AND. &
158      bp_keys_r(n,1) .EQ. rcut_in .AND. &

```

```

159      bp_keys_r(n,2) .EQ. alat_in) THEN
160
161      ! LOAD ID
162      bp_id = n
163
164      END IF
165      n = n + 1
166  END DO
167  calc_count = max(calc_count, bp_id)
168
169  IF(bp_id .EQ. 0)THEN
170
171      alat_temp = alat_in * maxval(uv_in)
172      uv_temp(1:3,1:3) = uv_in / maxval(uv_in)
173
174      ! Calculate number of eos points
175      eos_points = 2 * eos_size + 1
176
177      !# GET BPID
178      bp_configs_count = bp_configs_count + 1
179      bp_id = bp_configs_count
180
181      !# STORE UNIQUE DETAILS - LABEL, TYPE, EXPANSION, RCUT, ALAT
182      bp_keys_i(bp_id, 1) = label
183      bp_keys_i(bp_id, 2) = crystal_type
184      bp_keys_i(bp_id, 3) = expansion
185      bp_keys_r(bp_id,1) = rcut_in
186      bp_keys_r(bp_id,2) = alat_temp
187
188      !
189      known_atoms_per_crystal(bp_id) = atoms_per_crystal
190      known_expansion(bp_id) = expansion
191
192      !# CONFIG COUNTER FOR THIS BPID
193      cn_counter = 0
194
195      m = 0
196      DO i=1,expansion
197          DO j=1,expansion
198              DO k=1,expansion
199                  DO n=1,SIZE(primitive_labels, 1)
200                      m = m + 1
201                      expanded_labels(m) = primitive_labels(n)
202                      expanded_coords(m,1) = (i - 1 + primitive_coords(n,1)) / expansion
203                      expanded_coords(m,2) = (j - 1 + primitive_coords(n,2)) / expansion
204                      expanded_coords(m,3) = (k - 1 + primitive_coords(n,3)) / expansion
205                  END DO
206              END DO
207          END DO
208      END DO
209
210      !#####
211      ! EQUATION OF STATE
212      !#####
213
214      cc_start = cc + 1
215      cc_end = cc + eos_points
216      cc_log(bp_id, 1) = cc_start
217      cc_log(bp_id, 2) = cc_end
218
219

```

```

220
221 cc_counter = 1
222 rn = 1
223 bp_cn(bp_id, rn, 1) = cc_counter
224
225 DO n=1,eos_points
226   cc_counter = cc_counter + 1
227
228   ! Expand input uv
229   uv_config = expansion * uv_temp
230
231   ! Calculate strain
232   sigma = -(eos_size + 1) + n) * (eos_strain / eos_size)
233   strain(:, :) = 0.0D0
234   strain(1,1) = 1.0D0 + sigma
235   strain(2,2) = 1.0D0 + sigma
236   strain(3,3) = 1.0D0 + sigma
237
238   ! Apply strain
239   uv_config = matmul(strain, uv_config)
240
241   ! Calc volume
242   vol_cell(:, :) = alat_temp * uv_config(:, :)
243   vol = TripleProduct(vol_cell(1,:), vol_cell(2,:), vol_cell(3,:))
244
245 CALL add_config(rcut_in, alat_temp, uv_config, expanded_labels(1:m), expanded_coords(1:m,1:3)) ! Increments cc
246   by 1
247 cn_counter = cn_counter + 1
248 bp_configs(bp_id, cn_counter) = cc
249 CALL make_nl(cc)
250
251 !print *, bp_id, n, cc, rcut_in, alat_temp, m
252
253 ! RECORD DATA
254 calc_strains(bp_id, 1, n) = sigma
255 calc_volumes(bp_id, 1, n) = vol/m
256
257 ! REALS
258 bp_data_r(cc, 1) = rcut_in      ! rcut
259 bp_data_r(cc, 2) = alat_temp    ! alat
260 bp_data_r(cc, 3) = sigma        ! strain
261 bp_data_r(cc, 4) = vol          ! vol
262 bp_data_r(cc, 5) = vol/m       ! vol/atom
263 !bp_data_r(cc, 11) =           ! epa
264 !bp_data_r(cc, 12) =           ! epa_eos
265
266 ! INTS
267 bp_data_i(cc, 1) = n            ! eos number
268 bp_data_i(cc, 2) = cc_start     ! bp start
269 bp_data_i(cc, 3) = cc_end       ! bp end
270 bp_data_i(cc, 4) = m            ! atom count
271 !print *, key(cc, 1), key(cc, 2), key(cc, 3), key(cc, 4), key(cc, 5), key(cc, 6)
272
273 total_atoms = total_atoms + m
274 END DO
275 bp_cn(bp_id, rn, 2) = cc_counter - 1
276
277 !#####
278 ! ELASTIC CONSTANTS
279

```

```

280 !#####
281
282
283 ! Calculate number of eos points
284 ec_points = ec_size
285
286 DO ec_n = 1, 9
287   cc_start = cc + 1
288   cc_end = cc + ec_points
289   cc_log(bp_id, 2*ec_n+1) = cc_start
290   cc_log(bp_id, 2*ec_n+2) = cc_end
291
292   rn = rn + 1
293   bp_cn(bp_id, rn, 1) = cc_counter
294
295 DO n=1,ec_points
296   cc_counter = cc_counter + 1
297
298 ! Expand input uv
299 uv_config = expansion * uv_temp
300
301 ! Calculate strain
302 sigma = (n - 1) * (eos_strain / ec_points)
303 strain(:,:) = 0.0D0
304
305 IF(ec_n .EQ. 1)THEN
306   strain(1,1) = 1.0D0 + sigma
307   strain(2,2) = 1.0D0
308   strain(3,3) = 1.0D0
309 ELSE IF (ec_n .EQ. 2)THEN
310   strain(1,1) = 1.0D0
311   strain(2,2) = 1.0D0 + sigma
312   strain(3,3) = 1.0D0
313 ELSE IF (ec_n .EQ. 3)THEN
314   strain(1,1) = 1.0D0
315   strain(2,2) = 1.0D0
316   strain(3,3) = 1.0D0 + sigma
317 ELSE IF (ec_n .EQ. 4)THEN
318   strain(1,1) = 1.0D0 / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
319   strain(2,2) = 1.0D0 / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
320   strain(3,3) = 1.0D0 / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
321   strain(2,3) = sigma / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
322   strain(3,2) = sigma / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
323 ELSE IF (ec_n .EQ. 5)THEN
324   strain(1,1) = 1.0D0 / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
325   strain(2,2) = 1.0D0 / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
326   strain(3,3) = 1.0D0 / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
327   strain(1,3) = sigma / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
328   strain(3,1) = sigma / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
329 ELSE IF (ec_n .EQ. 6)THEN
330   strain(1,1) = 1.0D0 / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
331   strain(2,2) = 1.0D0 / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
332   strain(3,3) = 1.0D0 / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
333   strain(1,2) = sigma / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
334   strain(2,1) = sigma / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
335 ELSE IF (ec_n .EQ. 7)THEN
336   strain(1,1) = (1.0D0 + sigma) / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
337   strain(2,2) = (1.0D0 - sigma) / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
338   strain(3,3) = (1.0D0) / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
339 ELSE IF (ec_n .EQ. 8)THEN
340   strain(1,1) = (1.0D0 + sigma) / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))

```

```

341     strain(2,2) = (1.0D0) / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
342     strain(3,3) = (1.0D0 - sigma) / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
343 ELSE IF (ec_n .EQ. 9)THEN
344     strain(1,1) = (1.0D0) / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
345     strain(2,2) = (1.0D0 + sigma) / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
346     strain(3,3) = (1.0D0 - sigma) / ((1.0D0 - sigma**2)**(1.0D0/3.0D0))
347 END IF
348
349
350 ! Apply strain
351 uv_config = matmul(strain, uv_config)
352
353 ! Calc volume
354 vol_cell(:,:) = alat_temp * uv_config(:,:)
355 vol = TripleProduct(vol_cell(1,:), vol_cell(2,:), vol_cell(3,:))
356
357 CALL add_config(rcut_in, alat_temp, uv_config, expanded_labels(1:m), expanded_coords(1:m,1:3)) ! Increments
358             cc by 1
359 cn_counter = cn_counter + 1
360 bp_configs(bp_id, cn_counter) = cc
361 CALL make_nl(cc)
362
363 ! RECORD DATA
364 calc_strains(bp_id, ec_n + 1, n) = sigma
365 calc_volumes(bp_id, ec_n + 1, n) = vol/m
366
367 ! RECORD DATA
368 bp_data_r(cc, 1) = rcut_in          ! rcut
369 bp_data_r(cc, 2) = alat_temp        ! alat
370 bp_data_r(cc, 3) = sigma            ! strain
371 bp_data_r(cc, 4) = vol              ! vol
372 bp_data_r(cc, 5) = vol/m           ! vol/atom
373 !bp_data_r(cc, 11) =               ! epa
374 !bp_data_r(cc, 12) =               ! epa_eos
375
376 bp_data_i(cc, 1) = n                ! eos number
377 bp_data_i(cc, 2) = cc_start         ! bp start
378 bp_data_i(cc, 3) = cc_end           ! bp end
379 bp_data_i(cc, 4) = m                ! atom count
380 !print *, key(cc, 1), key(cc, 2), key(cc, 3), key(cc, 4), key(cc, 5), key(cc, 6)
381
382 total_atoms = total_atoms + m
383 END DO
384 bp_cn(bp_id, rn, 2) = cc_counter - 1
385
386 END DO
387
388 END IF
389
390
391 END SUBROUTINE add_inner

```

## O.3 Main fitting class

The fitting class is in Python and is given in listing ???. It loops over the required fitting settings provided by the user and attempts to optimise the parameters for the potentials.

Listing O.3: Potential fitting - main class (Python)

---

```

1 ######
2 # Ben Palmer University of Birmingham 2020
3 # Free to use
4 #####
5
6 import numpy
7 from scipy.optimize import minimize
8 from scipy.optimize import basinhopping
9 from potential import potential
10 from gd import gd
11 from rescale_density import rescale_density
12 from pot_fit_start import pf_start
13 from pot_fit_steps import pf_steps
14 from pot_fit_genetic import pf_genetic
15 from pot_fit_data import pf_data
16 from pot_fit_cycle import pf_cycle
17 from pot_fit_potential import pf_potential
18 from pot_fit_generation import pf_generation
19 from pot_fit_extinction import pf_extinction
20 from pot_fit_enhance import pf_enhance
21 from pot_fit_parameters import pf_parameters
22 from pot_fit_setop import pf_setop
23 from pot_fit_gd import pf_gd
24 from pot_fit_ng import pf_ng
25 from pot_fit_sa import pf_sa
26 from pot_fit_neldermead import pf_nm
27 from pot_fit_bh import pf_bh
28 from pot_fit_cg import pf_cg
29 from pot_fit_bfgs import pf_bfgs
30 from pot_fit_gsa import pf_gsa
31 from pot_fit_saa import pf_saa
32 from pot_fit_sps import pf_sps
33 from pot_fit_saved import pf_saved
34 from pot_fit_random import pf_random
35 from pot_fit_top import pf_top
36 from pot_fit_pgradient import pf_top
37 from pot_fit_display import pf_display
38 from pot_fit_display_simple import pf_display_simple
39 from pot_fit_init import pf_init
40 from pot_fit_save import pf_save
41 from pot_fit_final import pf_final
42 import matplotlib.pyplot as plt
43 import time
44 import random
45 import hashlib
46
47
48 class pf:
49
50     start_time = 0.0
51     end_time = 0.0
52     rss_plot_data = []
53
54     def run():
55
56         # Start Time
57         pf.start_time = time.time()
58
59         pf.pbest_dir = g.dirs['wd'] + '/fitting/saved_potentials/0000_pbest/'
60         std.make_dir(pf.pbest_dir)
61

```

```

62      # Log
63      main.log_title("Potential Fit")
64
65      # Set up EFS and BP modules and g.pfdata
66      pf_init.run()
67
68      # Create Plot Dirs
69      std.make_dir(g.dirs['wd'] + '/fitting')
70      pf.fh = open(g.dirs['wd'] + '/fitting/fitting_summary.txt', 'w')
71
72
73      # Run once with initial parameters
74      g.pfdata['stage'] = 'START - INITIAL PARAMETERS'
75      g.pfdata['stage_brief'] = 'START'
76      rss = pf.get_rss()
77
78
79      for fit in g.fit:
80          pf.fit = fit
81          if(pf.fit['type'].lower() == "random"):
82              pf_random.run(fit)
83          elif(pf.fit['type'].lower() == "sa"):
84              pf_sa.run(fit)
85          elif(pf.fit['type'].lower() == "gsa"):
86              pf_gsa.run(fit)
87          elif(pf.fit['type'].lower() == "ga"):
88              pf_genetic.run(fit)
89          elif(pf.fit['type'].lower() == "ng"):
90              pf_ng.run(fit)
91          elif(pf.fit['type'].lower() == "sps"):
92              pf_sps.run(fit)
93          elif(pf.fit['type'].lower() == "nm"):
94              pf_neldermead.run(fit)
95          elif(pf.fit['type'].lower() == "bfgs"):
96              pf_bfgs.run(fit)
97          elif(pf.fit['type'].lower() == "bh"):
98              pf_bh.run(fit)
99          elif(pf.fit['type'].lower() == "cg"):
100             pf_cg.run(fit)
101
102     pf_save.top("BEST_PARAMETERS")
103
104     potential.update(g.pfdata['p']['best'])
105     rss = pf.get_rss()
106
107     pf_top.save(g.dirs['wd'] + '/save', 'top_parameters.txt')
108
109     # Output
110     #potential_output.full()
111
112     # End Time
113     pf.end_time = time.time()
114
115     # Log fit time
116     main.log("Fit time: ", str(pf.end_time - pf.start_time))
117
118     pf_save.rss_plot_full()
119
120
121 ##########
122 # SET UP - initialise EFS, BP and any other modules

```

```

123     #         and run first calc
124 #####
125
126
127 def get_rss(save_in_top=True, quiet=False):
128     rss = rss_calc.run_calc(save_in_top)
129
130     g.pfdata['rss']['current'] = rss
131     g.pfdata['rss']['counter'] += 1
132
133     # If successful
134     if(rss is not None and not quiet):
135
136         # Store Details
137         g.pfdata['bp']['current'] = g.bp_results.copy()
138         g.pfdata['rss']['counter_successful'] += 1
139         g.pfdata['rss']['since_improvement'] += 1
140         if(g.pfdata['rss']['start'] == None):
141             g.pfdata['rss']['start'] = rss
142         if(g.pfdata['rss']['best'] == None or rss < g.pfdata['rss']['best']):
143             g.pfdata['p']['best'] = numpy.copy(potential.pot['p'])
144             g.pfdata['rss']['best'] = rss
145             g.pfdata['bp']['best'] = g.bp_results.copy()
146             g.pfdata['efs']['best'] = g.efs_results.copy()
147             g.pfdata['rss']['since_improvement'] = 0
148
149             potential.save(pf.pbest_dir, g.pfdata['p']['best'])
150             pf_save.save_summary(pf.pbest_dir, g.pfdata['p']['best'], g.pfdata['rss']['best'], g.pfdata['bp']['best'],
151                                 g.pfdata['bp']['known'], g.pfdata['efs']['best'], g.pfdata['efs']['known'])
152             potential.plot(pf.pbest_dir)
153
154         # Save in top
155         if(save_in_top):
156             list_len = g.fitting['top_parameters']
157
158             # COPY ARRAYS
159             p_now = numpy.copy(potential.pot['p'])
160             efs_results = g.efs_results.copy()
161             bp_results = g.bp_results.copy()
162             rss_details = g.rss.copy()
163
164             if(len(g.top_parameters) == 0):
165                 g.top_parameters.append([rss, p_now, efs_results, bp_results, bp.max_density, efs.max_density,
166                                         rss_details])
167             else:
168                 i = 0
169                 while(i<len(g.top_parameters)):
170                     if(rss < g.top_parameters[i][0]):
171                         break
172                     elif(rss == g.top_parameters[i][0] and g.top_parameters[i][1].all() == p_now.all()):
173                         i = list_len
174                         break
175                     else:
176                         i = i + 1
177                 if(i < list_len):
178                     g.top_parameters.insert(i, [rss, p_now, efs_results, bp_results, bp.max_density, efs.max_density,
179                                         rss_details])
180             if(len(g.top_parameters) > list_len):
181                 g.top_parameters.pop()
182
183         # DISPLAY

```

```

181     if(not quiet):
182         pf_display.output()
183 #for i in range(min(len(g.top_parameters),10)):
184 #    print(g.top_parameters[i][0])
185     return rss
186
187
188
189     def summary_line(opt_type, t_start, t_end, rss_start, rss_end):
190
191         t_taken = str('{:6.3f}'.format(t_end - t_start))
192         while(len(t_taken)<16):
193             t_taken = t_taken + " "
194
195         rss_start = str('{:12.4e}'.format(rss_start))
196         while(len(rss_start)<16):
197             rss_start = rss_start + " "
198
199         rss_end = str('{:12.4e}'.format(rss_end))
200         while(len(rss_end)<16):
201             rss_end = rss_end + " "
202
203         opt_type = str(opt_type)
204         while(len(opt_type)<30):
205             opt_type = opt_type + " "
206
207         pf.fh.write(opt_type + rss_start + "--> " + rss_end + " " + t_taken + "\n")
208
209
210     def save_rss_plot_data(t_start, name, data):
211         plot_data = numpy.asarray(data)
212         pf.rss_plot_data.append([t_start, name, data])

```

---

## O.4 Cubic Splines

Fortran computes the cubic spline in the fnc module and the subroutine for the Heaviside type spline is in listing ??.

Listing O.4: A cubic spline, using the Heaviside step function to cutoff each summed polynomial

```

1 ! Two Band Modelling Fe-Cr Olsson, Wallenius
2 ! CUBIC SPLINE
3 ! sum (ai (r - ri)^3 H(ri - r)
4 ! SCALAR SUBROUTINE
5 SUBROUTINE cubic_spline(r, p, p_fixed, y)
6 !#####
7 ! p coefficients
8 ! pf r cutoffs
9 ! they must be the same size
10 IMPLICIT NONE
11 !#####
12 REAL(kind=DoubleReal), INTENT(IN) :: r
13 REAL(kind=DoubleReal), INTENT(IN) :: p(:)
14 REAL(kind=DoubleReal), INTENT(IN) :: p_fixed(:)
15 REAL(kind=DoubleReal), INTENT(OUT) :: y
16 !#####
17 REAL(kind=DoubleReal) :: H
18 INTEGER(kind=StandardInteger) :: n
19 !#####

```

```

20  y = 0.0DO
21  DO n = 1, SIZE(p,1)
22    ! Heaviside((p_fixed(n) - r)
23    IF((p_fixed(n) - r) < 0.0DO)THEN
24      H = 0.0DO
25    ELSE
26      H = 1.0DO
27    END IF
28    y = y + p(n) * (r - p_fixed(n))**3 * H
29  END DO
30  IF(SIZE(p,1) + 1 .EQ. SIZE(p_fixed,1))THEN
31    IF(r .GE. p_fixed(SIZE(p_fixed,1)))THEN
32      y = 0.0DO
33    ELSE
34      y = y * (r - p_fixed(SIZE(p_fixed,1)))**3
35    END IF
36  END IF
37  END SUBROUTINE cubic_spline
38
39  ! VECTOR SUBROUTINE
40  SUBROUTINE cubic_spline_v(r, p, p_fixed, y)
41  !#####
42  ! CUBIC SPLINE
43  IMPLICIT NONE
44  !#####
45  REAL(kind=DoubleReal), INTENT(IN) :: r(:)
46  REAL(kind=DoubleReal), INTENT(IN) :: p(:)
47  REAL(kind=DoubleReal), INTENT(IN) :: p_fixed(:)
48  REAL(kind=DoubleReal), INTENT(OUT) :: y(1:SIZE(r,1))
49  INTEGER(kind=StandardInteger) :: n
50  !#####
51  ! Loop through all the values in r(:), calculate and store in y(:)
52  DO n = 1, SIZE(r,1)
53    CALL cubic_spline(r(n), p, p_fixed, y(n))
54  END DO
55  END SUBROUTINE cubic_spline_v

```

---

## O.5 Spline Subroutine for the Cubic Knot Spline

The cubic knot spline is also computed in Fortran (listing ??) and uses the solve linear set (sls\_solve) subroutine.

Listing O.5: A cubic knot spline that could be extended to a quintic.

```

1  SUBROUTINE spline_ab(spline_type, node_a, node_b, coeffs)
2  !#####
3  IMPLICIT NONE
4  !#####
5  INTEGER(kind=StandardInteger), INTENT(IN) :: spline_type
6  REAL(kind=DoubleReal), INTENT(IN) :: node_a(:)
7  REAL(kind=DoubleReal), INTENT(IN) :: node_b(:)
8  REAL(kind=DoubleReal), INTENT(OUT) :: coeffs(1:SIZE(node_a) + SIZE(node_b) - 2)
9  !#####
10 coeffs = 0.0DO
11 ! POLY3
12 ! Each node/knot has 3 values x, y, y'
13 IF(spline_type .EQ. 1 .AND. SIZE(node_a) .EQ. 3 .AND. SIZE(node_b) .EQ. 3)THEN
14   CALL spline_ab_poly(node_a, node_b, coeffs)
15 END IF
16 ! POLY5
17 ! Each node/knot has 4 values x, y, y', y''
```

```

18 IF(spline_type .EQ. 2 .AND. SIZE(node_a) .EQ. 4 .AND. SIZE(node_b) .EQ. 4)THEN
19   CALL spline_ab_poly(node_a, node_b, coeffs)
20 END IF
21 END SUBROUTINE spline_ab
22
23
24 SUBROUTINE spline_ab_poly(node_a, node_b, coeffs)
25 !#####
26 IMPLICIT NONE
27 !#####
28 REAL(kind=DoubleReal), DIMENSION(:), INTENT(IN) :: node_a
29 REAL(kind=DoubleReal), DIMENSION(:), INTENT(IN) :: node_b
30 REAL(kind=DoubleReal), INTENT(OUT) :: coeffs(1:SIZE(node_a) + SIZE(node_b) - 2)
31 !#####
32 INTEGER(kind=StandardInteger) :: m_size = 0
33 INTEGER(kind=StandardInteger) :: m_half_size = 0
34 INTEGER(kind=StandardInteger) :: d_loop = 0
35 REAL(kind=DoubleReal) :: x(1:SIZE(node_a,1)+SIZE(node_b,1)-2,1:SIZE(node_a,1)+SIZE(node_b,1)-2)
36 REAL(kind=DoubleReal) :: y(1:SIZE(node_a,1)+SIZE(node_b,1)-2)
37 REAL(kind=DoubleReal) :: x_coeffs(1:(SIZE(node_a,1)+SIZE(node_b,1)-2))
38 REAL(kind=DoubleReal) :: x_exponent(1:(SIZE(node_a,1)+SIZE(node_b,1)-2))
39 INTEGER(kind=StandardInteger) :: n, row, col, a_row
40 !#####
41 ! Sizes
42 m_size = SIZE(node_a) + SIZE(node_b) - 2
43 m_half_size = m_size / 2
44 ! x exponents
45 DO n = 1, m_size
46   x_exponent(n) = 1.0D0 * (n - 1)
47 END DO
48 ! Coeffs
49 x_coeffs = 1
50 ! Make y matrix
51 row = 1
52 DO n=2,SIZE(node_a)
53   y(row) = node_a(n)
54   row = row + 1
55   y(row) = node_b(n)
56   row = row + 1
57 END DO
58 ! Make x matrix
59 row = 0
60 DO a_row = 1, m_half_size
61   ! NODE A
62   row = row + 1
63   DO col = 1, a_row -1
64     x(row, col) = 0.0D0
65   END DO
66   DO col = a_row, m_size
67     x(row, col) = x_coeffs(col) * node_a(1)**x_exponent(col)
68   END DO
69   ! NODE B
70   row = row + 1
71   DO col = 1, a_row -1
72     x(row, col) = 0.0D0
73   END DO
74   DO col = a_row, m_size
75     x(row, col) = x_coeffs(col) * node_b(1)**x_exponent(col)
76   END DO
77   ! UPDATE COEFFS AND EXPONENT
78   DO col = 1, m_size

```

```

79      x_coeffs(col) = x_coeffs(col) * x_exponent(col)
80  END DO
81  DO col = 1, m_size
82    IF(x_exponent(col) .GT. 0.0D0)THEN
83      x_exponent(col) = x_exponent(col) - 1.0D0
84    END IF
85  END DO
86 ! SOLVE
87 CALL sis_solve(x, y, coeffs)
88 !#####
89 !#####
90 END SUBROUTINE spline_ab_poly

```

---

## O.6 ZBL with Cubic Knot Spline

A further set of subroutines are used to compute the cubic spline summed to the ZBL function (listing ??).

Listing O.6: ZBL with cubic knot spline, variable x, y(x) and y'(x) parameters

```

1 SUBROUTINE cubic_knot_spline_5(r, p, p_fixed, y)
2 !#####
3 ! p coefficients
4 ! pf r cutoffs
5 ! they must be the same size
6 IMPLICIT NONE
7 !#####
8 REAL(kind=DoubleReal), INTENT(IN) :: r
9 REAL(kind=DoubleReal), INTENT(IN) :: p(:)
10 REAL(kind=DoubleReal), INTENT(IN) :: p_fixed(:)
11 REAL(kind=DoubleReal), INTENT(OUT) :: y
12 !#####
13 INTEGER(kind=StandardInteger) :: n
14 REAL(kind=DoubleReal) :: nodes(1:(size(p,1)-2) / 3 + 2,1:4)
15 REAL(kind=DoubleReal) :: qa, qb, zbl_mult
16 INTEGER(kind=StandardInteger) :: psize, pfszie, node_count, pstride
17 LOGICAL :: complete
18 !#####
19 psize = SIZE(p, 1)
20 pfszie = SIZE(p_fixed, 1)
21 pstride = (size(p,1)-2) / 3
22 node_count = pstride + 2
23 nodes(:, :) = 0.0D0
24 nodes(2:node_count-1, 1) = p(1:pstride)
25 nodes(2:node_count-1, 2) = p(pstride+1:2*pstride)
26 nodes(2:node_count-1, 3) = p(2*pstride+1:3*pstride)
27
28 qa = p_fixed(1)
29 qb = p_fixed(2)
30 zbl_mult = p_fixed(3)
31 nodes(1, 1) = p_fixed(4)
32 nodes(1, 2) = p(3*pstride+1)
33 nodes(1, 3) = p(3*pstride+2)
34 nodes(node_count, 1) = p_fixed(5)
35 nodes(node_count, 2) = p_fixed(6)
36 nodes(node_count, 3) = p_fixed(7)
37
38 IF(r .LT. nodes(1, 1))THEN
39   y = nodes(1, 2)
40 ELSE IF(r .GT. nodes(node_count, 1))THEN
41   y = nodes(node_count, 2)

```

```

42 ELSE
43 ! Add ZBL
44 y = y + 1.0D0 * f_zbl_ackson_mendelev(r, qa, qb)
45
46 ! IF EXACTLY A NODE
47 complete = .FALSE.
48 DO n = 1, node_count
49   IF(r .EQ. nodes(n,1))THEN
50     complete = .TRUE.
51     y = y + nodes(n,2)
52   END IF
53 END DO
54
55 IF(complete .EQV. .FALSE.)THEN
56   y = y + f_cubic_knot_spline(r, nodes)
57 END IF
58 END IF
59
60
61 END SUBROUTINE cubic_knot_spline_5
62
63
64 ! VECTOR SUBROUTINE
65 SUBROUTINE cubic_knot_spline_5_v(r, p, p_fixed, y)
66 !#####
67 ! PAIR SPLINE
68 IMPLICIT NONE
69 !#####
70 REAL(kind=DoubleReal), INTENT(IN) :: r(:)
71 REAL(kind=DoubleReal), INTENT(IN) :: p(:)
72 REAL(kind=DoubleReal), INTENT(IN) :: p_fixed(:)
73 REAL(kind=DoubleReal), INTENT(OUT) :: y(1:SIZE(r,1))
74 INTEGER(kind=StandardInteger) :: n
75 !#####
76 ! Loop through all the values in r(:), calculate and store in y(:)
77 DO n = 1, SIZE(r,1)
78   CALL cubic_knot_spline_5(r(n), p, p_fixed, y(n))
79 END DO
80 END SUBROUTINE cubic_knot_spline_5_v
81
82
83
84 FUNCTION f_cubic_knot_spline(r, nodes) RESULT (y)
85 !#####
86 REAL(kind=DoubleReal) :: r
87 REAL(kind=DoubleReal) :: nodes(:, :)
88 REAL(kind=DoubleReal) :: y
89 !#####
90 INTEGER(kind=StandardInteger) :: n, j
91 !#####
92
93 ! FIND n, n+1 TO SPLINE BETWEEN
94 IF(r .LT. nodes(1,1))THEN
95   n = 1
96 ELSE IF(r .GT. nodes(SIZE(nodes,1), 1))THEN
97   n = SIZE(nodes,1) - 1
98 ELSE
99   DO n = 1, SIZE(nodes,1) - 1
100     IF(r .GE. nodes(n, 1) .AND. r .LT. nodes(n+1, 1))THEN
101       EXIT
102     END IF

```

```

103      END DO
104  END IF
105  y = f_cubic_knot_ab(r, nodes(n, 1), nodes(n, 2), nodes(n, 3), nodes(n+1, 1), nodes(n+1, 2), nodes(n+1, 3))
106 END FUNCTION f_cubic_knot_spline
107
108
109 FUNCTION f_cubic_knot_ab(r, xa, ya, ypa, xb, yb, ypb) RESULT (y)
110 ! CUBIC KNOT
111 IMPLICIT NONE
112 !#####
113 REAL(kind=DoubleReal) :: r
114 REAL(kind=DoubleReal) :: xa, ya, ypa, xb, yb, ypb
115 REAL(kind=DoubleReal) :: y
116 !#####
117 REAL(kind=DoubleReal) :: xmat(1:4,1:4)
118 REAL(kind=DoubleReal) :: xmat_inv(1:4,1:4)
119 REAL(kind=DoubleReal) :: ymat(1:4)
120 REAL(kind=DoubleReal) :: c(1:4)
121 !#####
122 xmat(1,1) = 1.0
123 xmat(1,2) = xa
124 xmat(1,3) = xa**2
125 xmat(1,4) = xa**3
126 xmat(2,1) = 1.0
127 xmat(2,2) = xb
128 xmat(2,3) = xb**2
129 xmat(2,4) = xb**3
130 xmat(3,1) = 0.0
131 xmat(3,2) = 1.0D0
132 xmat(3,3) = 2.0D0 * xa
133 xmat(3,4) = 3.0D0 * xa**2
134 xmat(4,1) = 0.0
135 xmat(4,2) = 1.0D0
136 xmat(4,3) = 2.0D0 * xb
137 xmat(4,4) = 3.0D0 * xb**2
138
139 ymat(1) = ya
140 ymat(2) = yb
141 ymat(3) = ypa
142 ymat(4) = ypb
143
144 !CALL sls_solve(xmat, ymat, c)
145 CALL minverse(xmat, xmat_inv, 4)
146 c = MATMUL(xmat_inv, ymat)
147 y = c(1) + c(2) * r + c(3) * r**2 + c(4) * r**3
148
149 END FUNCTION f_cubic_knot_ab
150
151
152 FUNCTION f_zbl_ackland_mendelev(r, q1, q2) RESULT (y)
153 !#####
154 ! ZBL Development of an interatomic potential for phosphorus impurities in alpha-iron 2004
155 IMPLICIT NONE
156 !#####
157 REAL(kind=DoubleReal) :: q1, q2, r
158 REAL(kind=DoubleReal) :: y
159 !#####
160 REAL(kind=DoubleReal) :: rs, x, e
161 !#####
162 rs = 0.4683766D0 / (q1**((2.0D0/3.0D0) + q2**((2.0D0/3.0D0)))
163 x = (r / rs)

```

---

```

164 e = 0.1818D0 * exp(-3.2 * x) + 0.5099D0 * exp(-0.9423 * x) + 0.2802D0 * exp(-0.4029 * x) + 0.02817D0 * exp
165 (-0.2016 * x)
166 y = (q1 * q2 * e) / r
167 END FUNCTION f_zbl_ackland_mendelev

```

---

## O.7 Simulated Annealing

The simulated annealing algorithm is used in Python and is one of the global optimization algorithms (listing 5.3).

Listing O.7: Simulated Annealing global optimization function

---

```

1 """
2 Simple simulated annealing
3 No direction
4 Temperature loop from start to end temperature
5 Step size decreases with temperature
6
7 """
8
9 class pf_sa:
10
11     t = None
12     count = 0
13
14     def run(fit):
15         for repeat in range(fit['repeat']):
16             pf_sa.run_sa(fit)
17
18     def run_sa(fit):
19         pf_sa.count = pf_sa.count + 1
20         t_start = time.time()
21         start_best_rss = g.pfdata['rss']['best']
22
23         if(pf.fit['sa_loops_t'] == 0 or pf.fit['sa_loops_i'] == 0):
24             return 0
25
26         g.pfdata['stage'] = 'Simulated Annealing ' + str(pf_sa.count)
27         g.pfdata['stage_brief'] = 'SA' + str(pf_sa.count)
28
29         # Start - use best
30         p = g.pfdata['p']['best']
31         potential.update(p)
32         rss = pf.get_rss(False)
33         p_count = len(p)
34
35         # Store Best Parameters
36         p_best = numpy.copy(p)
37         rss_best = rss
38
39         rss_plot = []
40         rss_plot.append([time.time()-t_start, rss_best])
41
42         for tn in range(pf.fit['sa_loops_t']):
43             pv = pf.fit['sa_var'] * pf.fit['sa_var_factor']**tn
44             if(pf.fit['sa_loops_t'] == 1):
45                 pf_sa.t = pf.fit['sa_temp_start']
46             else:

```

---

```

48     pf_sa.t = numpy.round(pf.fit['sa_temp_start'] - tn * (pf.fit['sa_temp_start'] - pf.fit['sa_temp_end']) / (
49         pf.fit['sa_loops_t'] - 1), 6)
50     g.pfdata['stage'] = 'Simulated Annealing ' + str(pf_sa.count) + ' Loop ' + str(tn + 1) + ' Temp: ' + str(
51         pf_sa.t) + ' Pvar: ' + str(pv)
52     rss_start_loop = rss_best
53     for n in range(pf.fit['sa_loops_i']):
54         loop = True
55         while(loop):
56             p_new = potential.random(p, pv, False)
57             rss_new = pf.get_rss(False)
58             if(rss_new is not None):
59                 loop = False
60             if(rss_new is not None):
61                 if(rss_new < rss or numpy.random.uniform() < numpy.exp((rss-rss_new) / pf_sa.t)):
62                     p = numpy.copy(p_new)
63                     rss = rss_new
64                     if(rss_new < rss_best):
65                         p_best = numpy.copy(p_new)
66                         rss_best = rss_new
67                         rss_plot.append([time.time()-t_start, rss_best])
68                     if(rss_best < rss_start_loop):
69                         potential.update(p_best)
70                         pf.get_rss(True)
71                         p = numpy.copy(p_best)
72                         rss_plot.append([time.time()-t_start, rss_best])
73
74             pf.summary_line("SIMULATED ANNEALING " + str(pf_sa.count), t_start, time.time(), start_best_rss, g.pfdata['rss
75                 '][])
76             pf_save.top("SIMULATED_ANNEALING_" + str(pf_sa.count))
77             pf_save.rss_plot("SIMULATED_ANNEALING_" + str(pf_sa.count), rss_plot)
78             pf.save_rss_plot_data(t_start, "SIMULATED_ANNEALING_" + str(pf_sa.count), rss_plot)

```

## O.8 Genetic Algorithm

Another global optimization algorithm used in the Python section of the program is the genetic algorithm (listing ??).

Listing O.8: Genetic Algorithm global optimization function

```

1 #####
2 # Ben Palmer University of Birmingham 2020
3 # Free to use
4 #####
5
6 import numpy
7 from potential import potential
8 from display import display
9 from gd import gd
10 from rescale_density import rescale_density
11 from pot_fit_data import pf_data
12 from pot_fit_cycle import pf_cycle
13 from pot_fit_potential import pf_potential
14 from pot_fit_extinction import pf_extinction
15 from pot_fit_enhance import pf_enhance
16 from pot_fit_parameters import pf_parameters
17 import matplotlib.pyplot as plt
18 import time
19 import random

```

```

20 import hashlib
21
22
23 class pf_genetic:
24
25     count = 0
26     start_time = 0
27     variation_factor = 1.0
28
29     def run(fit):
30         if(fit['gens'] == 0):
31             return 0
32
33         #
34         t_start = time.time()
35         start_best_rss = g.pfdata['rss']['best']
36         pf_genetic.start_time = time.time()
37         pf_genetic.count = pf_genetic.count + 1
38         pf_genetic.rss_plot = []
39         pf_genetic.rss_plot.append([time.time() - pf_genetic.start_time, start_best_rss])
40         pf_genetic.rss_best = start_best_rss
41
42         g.pfdata['stage'] = 'Genetic Algorithm ' + str(pf_genetic.count)
43         g.pfdata['stage_brief'] = 'GA' + str(pf_genetic.count)
44
45         main.log_title("Genetic Fit")
46
47         # Load from input
48         pf_genetic.width = g.pfdata['psize']
49         pf_genetic.pop_size = fit['pop_size']
50         pf_genetic.fresh_size = fit['fresh_size']
51         pf_genetic.generations = fit['gens']
52         pf_genetic.no_clone_var = fit['no_clone_var']
53         pf_genetic.gen_variation_multiplier = fit['gen_variation_multiplier']
54
55
56
57         # Force to be even
58         if(pf_genetic.pop_size % 2 != 0):
59             pf_genetic.pop_size = pf_genetic.pop_size + 1
60         if(pf_genetic.fresh_size % 2 != 0):
61             pf_genetic.fresh_size = pf_genetic.fresh_size + 1
62
63         # Calculate other sizes
64         pf_genetic.children_size = pf_genetic.pop_size + 2 * pf_genetic.fresh_size
65         pf_genetic.merge_size = 2 * pf_genetic.pop_size + 3 * pf_genetic.fresh_size
66
67         # Set Up Pop + Fresh arrays
68         pf_genetic.pop = numpy.zeros((pf_genetic.pop_size, pf_genetic.width,),)
69         pf_genetic.pop_rss = numpy.zeros((pf_genetic.pop_size,),)
70         pf_genetic.fresh = numpy.zeros((pf_genetic.fresh_size, pf_genetic.width,),)
71         pf_genetic.fresh_rss = numpy.zeros((pf_genetic.fresh_size,),)
72         pf_genetic.children = numpy.zeros((pf_genetic.children_size, pf_genetic.width,),)
73         pf_genetic.children_rss = numpy.zeros((pf_genetic.children_size,),)
74         pf_genetic.pop_merged = numpy.zeros((pf_genetic.merge_size, pf_genetic.width,),)
75         pf_genetic.pop_merged_rss = numpy.zeros((pf_genetic.merge_size,),)
76
77         # Parents array
78         pf_genetic.parents = numpy.arange(pf_genetic.pop_size)
79
80         # Initialise population

```

```

81     g.pfdata['stage'] = 'Genetic Fit ' + str(pf_genetic.count) + ' Initialising'
82
83     # Copy out parameters from "top_parameters"
84     start_parameters = []
85     if(len(g.top_parameters) > 0):
86         for i in range(len(g.top_parameters)):
87             new_p = numpy.copy(g.top_parameters[i][1][:])
88             start_parameters.append(new_p)
89     else:
90         p = potential.get_start_parameters()
91         new_p = numpy.copy(p)
92         start_parameters.append(new_p)
93
94     # Load start parameters into pop, and fill remaining with random parameters
95     n = 0
96     for pn in range(pf_genetic.pop_size):
97         loop = True
98         while(loop):
99             n = n + 1
100            if(n <= len(start_parameters)):
101                pf_genetic.pop[pn, :] = start_parameters[n-1] [:]
102            else:
103                pf_genetic.pop[pn, :] = potential.random(g.pfdata['p']['best'], 1.0, False)
104            # Try - if it fails or rss == None, try next
105            potential.update(pf_genetic.pop[pn, :])
106            try:
107                # Update
108                rss = pf.get_rss()
109                if(rss is not None):
110                    loop = False
111                    pf_genetic.pop_rss[pn] = rss
112                except:
113                    pass
114                if(rss < pf_genetic.rss_best):
115                    pf_genetic.rss_best = rss
116                    pf_genetic.rss_plot.append([time.time() - pf_genetic.start_time, pf_genetic.rss_best])
117
118 ##########
119     # LOOP THROUGH GENERATIONS
120     #####
121
122     pf_genetic.generation = 0
123     for gen in range(pf_genetic.generations):
124         pf_genetic.generation = pf_genetic.generation + 1
125         pf_genetic.run_gen()
126
127     pf_genetic.rss_plot.append([time.time() - pf_genetic.start_time, pf_genetic.rss_best])
128
129     # End
130     pf.summary_line("GENETIC ALGORITHM " + str(pf_genetic.count), t_start, time.time(), start_best_rss, g.pfdata['rss']['best'])
131     pf_save.top("GENETIC_ALGORITHM_" + str(pf_genetic.count))
132     pf_save.rss_plot("GENETIC_ALGORITHM_" + str(pf_genetic.count), pf_genetic.rss_plot)
133     pf.save_rss_plot_data(t_start, "GENETIC_ALGORITHM_" + str(pf_genetic.count), pf_genetic.rss_plot)
134
135
136
137     def run_gen():
138
139         ss = 'Genetic Fit ' + str(pf_genetic.count) + ' Gen ' + str(pf_genetic.generation) + ' '

```

```

141     # Shuffle parents array
142     numpy.random.shuffle(pf_genetic.parents)
143
144     c = 0
145     # Breed Population
146     g.pfdata['stage'] = ss + 'Breed Population'
147     for p in range(pf_genetic.pop_size // 2):
148         loop = True
149         while(loop):
150             loop = pf_genetic.breed_event(p, c, 'p+p')
151             c = c + 2
152
153     # Make Fresh
154     g.pfdata['stage'] = ss + 'Initialising Fresh Population'
155     pf_genetic.make_fresh()
156
157
158     # Shuffle parents array
159     numpy.random.shuffle(pf_genetic.parents)
160
161     # Breed Population-Fresh
162     g.pfdata['stage'] = ss + 'Breed With Fresh Population'
163     for p in range(pf_genetic.fresh_size):
164         loop = True
165         while(loop):
166             loop = pf_genetic.breed_event(p, c, 'p+f')
167             c = c + 2
168
169     # Merge populations and select top to form next generation
170     g.pfdata['stage'] = ss + 'Merge Populations'
171     pf_genetic.merge()
172
173
174     # Load best back into population
175     pf_genetic.pop[:, :] = pf_genetic.pop_merged[0:pf_genetic.pop_size, :]
176     pf_genetic.pop_rss[:] = pf_genetic.pop_merged_rss[0:pf_genetic.pop_size]
177
178     # Change variation for next generation
179     pf_genetic.variation_factor = pf_genetic.variation_factor * pf_genetic.gen_variation_multiplier
180
181
182
183
184     def breed_event(p, c, opt):
185         pc = pf_genetic.width
186
187         pa = pf_genetic.parents[p]
188         if(opt == 'p+p'):
189             pb = pf_genetic.parents[p + pf_genetic.pop_size // 2]
190             pb_array = 'pop'
191         if(opt == 'p+f'):
192             pb = p
193             pb_array = 'fresh'
194
195         ca = c
196         cb = c + 1
197
198         # Breed
199         state = pf_genetic.get_state()
200         for i in range(pc):

```

```

202     state = pf_genetic.get_state(state)
203     if(state):
204         pf_genetic.children[ca, i] = pf_genetic.pop[pa, i]
205         if(pb_array == 'pop'):
206             pf_genetic.children[cb, i] = pf_genetic.pop[pb, i]
207         elif(pb_array == 'fresh'):
208             pf_genetic.children[cb, i] = pf_genetic.fresh[pb, i]
209     else:
210         if(pb_array == 'pop'):
211             pf_genetic.children[ca, i] = pf_genetic.pop[pb, i]
212         elif(pb_array == 'fresh'):
213             pf_genetic.children[ca, i] = pf_genetic.fresh[pb, i]
214     pf_genetic.children[cb, i] = pf_genetic.pop[pa, i]
215
216
217 # No clones - Child A
218 for pn in range(pf_genetic.pop_size):
219     if((pf_genetic.children[ca, :].all() == pf_genetic.pop[pn,:]).all()):
220         r = numpy.random.rand(pc)
221         pf_genetic.children[ca, :] = pf_parameters(pf_genetic.children[ca, :], pf_genetic.no_clone_var)
222         break
223
224 # No clones - Child B
225 for pn in range(pf_genetic.pop_size):
226     if((pf_genetic.children[cb, :].all() == pf_genetic.pop[pn,:]).all()):
227         r = numpy.random.rand(pc)
228         pf_genetic.children[cb, :] = pf_parameters(pf_genetic.children[cb, :], pf_genetic.no_clone_var)
229         break
230
231 # Check the children actually give valid parameters
232 loop = False
233 potential.update(pf_genetic.children[ca, :])
234 rss_a = pf.get_rss()
235 if(rss_a is None):
236     loop = True
237
238 potential.update(pf_genetic.children[cb, :])
239 rss_b = pf.get_rss()
240 if(rss_b is None):
241     loop = True
242
243 if(loop == False):
244     pf_genetic.children_rss[ca] = rss_a
245     pf_genetic.children_rss[cb] = rss_b
246
247 if(rss_a != None and rss_b != None):
248     if(rss_a < pf_genetic.rss_best):
249         pf_genetic.rss_best = rss_a
250         pf_genetic.rss_plot.append([time.time() - pf_genetic.start_time, pf_genetic.rss_best])
251     if(rss_b < pf_genetic.rss_best):
252         pf_genetic.rss_best = rss_b
253         pf_genetic.rss_plot.append([time.time() - pf_genetic.start_time, pf_genetic.rss_best])
254
255
256 # Loop again or not
257 return loop
258
259
260
261 def get_state(state = None):
262     if(state == None):

```

```

263     if(random.uniform(0.0, 100.0) > 50.0):
264         return True
265     return False
266 else:
267     if(random.uniform(0.0, 100.0) > 50.0):
268         if(state):
269             return False
270         return True
271     return state
272
273
274
275 def make_fresh():
276     for pn in range(pf_genetic.fresh_size):
277         loop = True
278         pbest = numpy.copy(g.top_parameters[0][1])
279         while(loop):
280             try:
281                 rn = int(min(len(g.top_parameters) - 1, numpy.floor((len(g.top_parameters) + 1) * random.uniform(0.0,
282                                         1.0)**3)))
283                 p_trial = numpy.copy(g.top_parameters[rn][1])
284                 p_new = potential.random(p_trial, pf_genetic.variation_factor, False)
285                 rss = pf.get_rss()
286                 if(rss is not None):
287                     loop = False
288             except:
289                 pass
290             pf_genetic.fresh[pn, :] = p_new
291             pf_genetic.fresh_rss[pn] = rss
292
293             if(rss < pf_genetic.rss_best):
294                 pf_genetic.rss_best = rss
295                 pf_genetic.rss_plot.append([time.time() - pf_genetic.start_time, pf_genetic.rss_best])
296
297 # Used to merge parents, fresh and all children into ordered array
298 def merge():
299     ps = pf_genetic.pop_size
300     fs = pf_genetic.fresh_size
301     cs = pf_genetic.pop_size + 2 * pf_genetic.fresh_size
302
303     # Reset array
304     pf_genetic.pop_merged[:, :] = 0.0
305     pf_genetic.pop_merged[0:ps, :] = pf_genetic.pop[:, :]
306     pf_genetic.pop_merged[ps:ps+fs, :] = pf_genetic.fresh[:, :]
307     pf_genetic.pop_merged[ps+fs:ps+fs+cs, :] = pf_genetic.children[:, :]
308     pf_genetic.pop_merged_rss[:] = 0.0
309     pf_genetic.pop_merged_rss[0:ps] = pf_genetic.pop_rss[:]
310     pf_genetic.pop_merged_rss[ps:ps+fs] = pf_genetic.fresh_rss[:]
311     pf_genetic.pop_merged_rss[ps+fs:ps+fs+cs] = pf_genetic.children_rss[:]
312
313     # Sort
314     pf_genetic.sort_merged()
315
316 def sort_merged():
317     sort.sort_1d_dp_asc(pf_genetic.pop_merged_rss[:])
318     pf_genetic.pop_merged_rss = sort.apply_keytable_1d_dp(pf_genetic.pop_merged_rss)
319     pf_genetic.pop_merged = sort.apply_keytable_2d_dp(pf_genetic.pop_merged)

```

## O.9 Gradient Descent

Gradient descent using a line search is one option to refine the genetic algorithm (listing ??), but on reflection it may have been better to use the algorithms in SciPy to enhance members of the genetic algorithm population.

Listing O.9: Gradient descent pseudocode

---

```

1 // Simulated Annealing
2 subroutine simulated_annealing(f, p, p_best)
3
4 // f is the function being optimised
5 // p is an array of the parameters
6
7 p_search = True
8 rss_best = calc_rss(f, p)
9
10 do while(p_search)
11
12     // Some function to calculate the
13     df(:) = gradient(f,p(:))
14
15     // Back track line search
16     alpha = 1.0
17     alpha_search = True
18     last_rss = -1.0
19     do while(alpha_search)
20         rss = calc_rss(f, p(:) - alpha * df(:))
21
22         // If it's the first step, store rss into last_rss
23         if(last_rss == -1.0)then
24             last_rss = rss
25         end if
26
27         // If the new one is worse, stop searching and step alpha back
28         if(rss > ls_rss)then
29             alpha_search = False
30             alpha = 2.0 * alpha
31             // If it's better, reduce alpha and keep searching
32             else
33                 alpha = 0.5 * alpha
34                 last_rss = rss
35             end if
36         end do
37
38     // Check new parameters
39     rss = calc_rss(f, p(:) - alpha * df(:))
40
41     // If it's better, save and keep searching
42     if(rss < rss_best)then
43         p(:) = p(:) - alpha * df(:)
44         rss_best = rss
45
46     // Break out
47     else
48         p_search = False
49     end if
50 end do
51
52
53 end subroutine

```

---

## O.10 Interpolation

Lagrange polynomial interpolation was programmed in Fortran and was used in a number of places throughout the program (listing ??).

Listing O.10: Lagrange polynomial interpolation

---

```

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

```

---

## O.11 Interpolation (Gradient)

A modified version of the Lagrange polynomial interpolation was programmed in Fortran to interpolate the derivative of a set of data points at a given position between (and including) the start and end point (listing ??).

Listing O.11: Lagrange polynomial interpolation (Gradients)

---

```

1 function lpinterp_dydx(x, data)
2 // x - point value being interpolated at fo calc f'(x)
3 // data - four x,y data points (n by 2 array)
4 n = len(data, 1)
5 dydx = 0.0
6 for i = 1,n
7   fx = 1.0
8   gx = 1.0
9   for j = 1, n
10    if (i != j) then
11      fx = fx / (data[i][1] - data[j][1])
12      psum = 1.0
13      for k = 1, n
14        if (i != k and j != k) then
15          psum = psum * (x - data[k][1])
16        end if
17      next k
18      gx = gx + psum
19    end if
20  next j
21  dydx = dydx + fx * gx * data[i][2]
22 next i
23 end function lpinterp_dydx

```

---

# Appendix P

## Potential Functions

### P.1 Introduction

This section of the appendix covers the types of potential function and common choices of function used and details on how to use these in the potential fitting code.

### P.2 Functions

#### P.2.1 Ackland-Mendelev Pair

$$v(r) = \sum_{i=1}^N (a_i(r - r_i)^3 H(r_i - r)) \quad (\text{P.1})$$

Listing P.1: Ackland Mendelev Pair

---

```
1 #TYPE ackland_mendelev_pair
2 #P 1.0
3 #PF 0.0
```

---

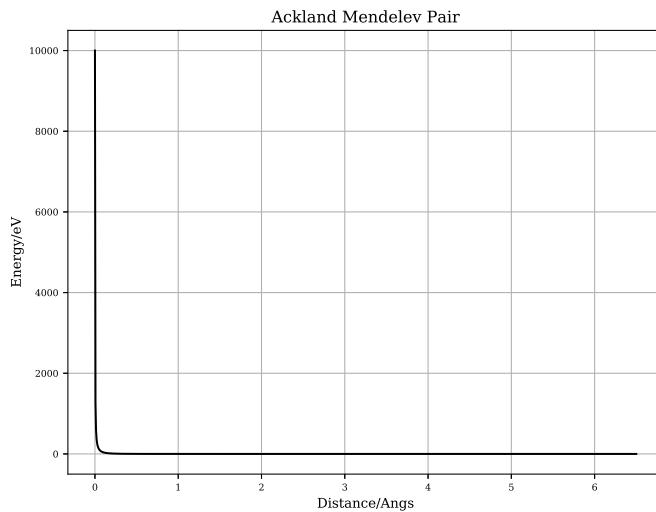


Figure P.1: Ackland Mendelev Pair

## P.2.2 Buckingham

$$V(r) = A \times \exp(-B \times r) - \frac{C}{r^6} \quad (\text{P.2})$$

Listing P.2: Buckingham

---

```
1 #TYPE buckingham
2 #P 6.0 0.5 12.0
```

---

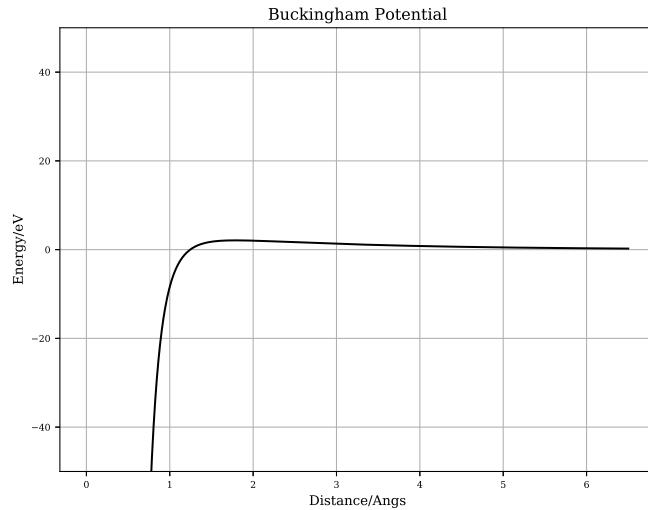


Figure P.2: Buckingham Potential

### P.2.3 Cubic Knot Spline

$$V(r) = \sum_i^N a_i (r - r_i)^3 H(r_i - r)$$

where

$$H(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} \quad (P.3)$$

Listing P.3: Cubic Knot Spline

---

```
1 #TYPE cubic_knot_spline
2 #P 0.0 1.0 0.02 -0.01 0.002 0.0
3 #PF 0.0 1.0 2.0 3.0 4.0 5.0
```

---

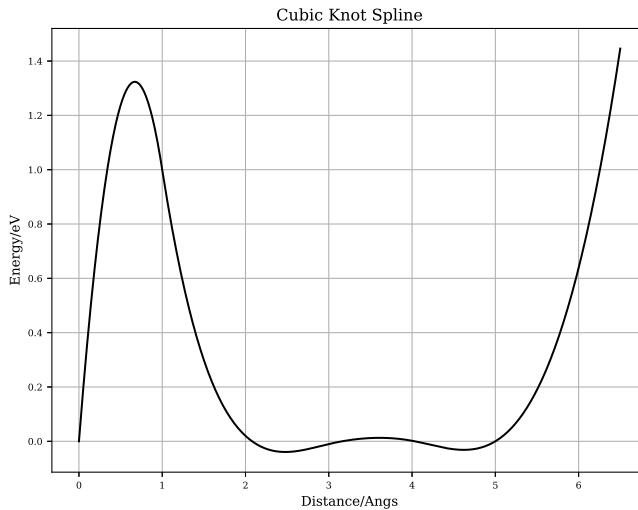


Figure P.3: Cubic Knot Spline

### P.2.4 Cubic Knot Spline Fixed End

$$V(r) = \sum_i^N a_i (r - r_i)^3 H(r_i - r)$$

where

$$H(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} \quad (P.4)$$

Listing P.4: Cubic Knot Spline Fixed End

---

```
1 #TYPE cubic_knot_spline_fixed_end
2 #P 0.0 1.0 0.02 -0.01 0.002 0.0
3 #PF 0.0 1.0 2.0 3.0 4.0 5.0 6.5 0.0 0.0
```

---

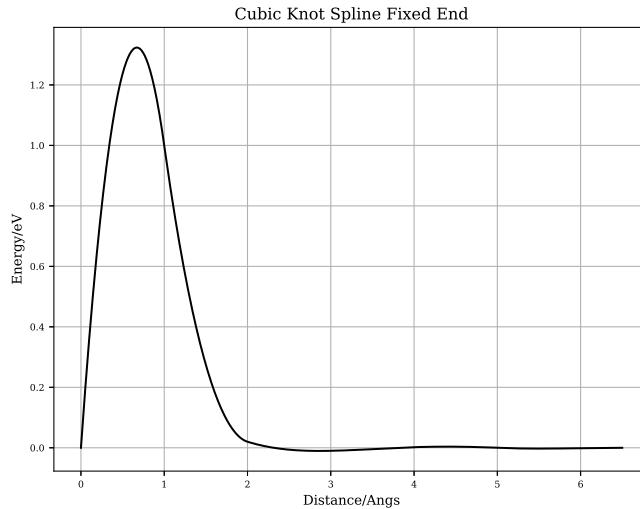


Figure P.4: Cubic Knot Spline Fixed End

### P.2.5 Cubic Knot Spline Fixed End Pair

$$V(r) = \sum_i^N a_i (r - r_i)^3 H(r_i - r)$$

where

$$H(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} \quad (P.5)$$

Listing P.5: Cubic Knot Spline Fixed End Pair

---

```

1 #TYPE cubic_knot_spline_fixed_end_pair
2 #P 0.5 0.007 -0.3 0.000001 0.002 0.0
3 #PF 2.0 2.5 3.0 3.5 4.0 5.0 26.0 26.0 1.0 6.5 0.0 0.0

```

---

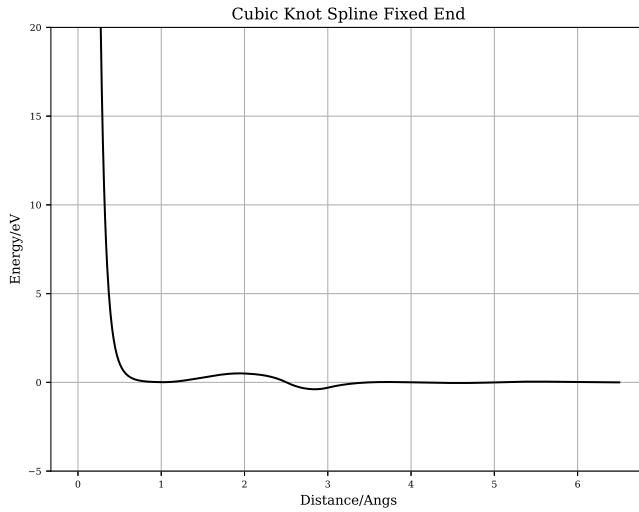


Figure P.5: Cubic Knot Spline Fixed End Pair

## P.2.6 Cubic Splines

$$V(r) = \sum_i^N a_i (r - r_i)^3 H(r_i - r)$$

where

$$H(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} \quad (\text{P.6})$$

This function requires two sets of parameters. P is a list of N coefficients and PF is a list of N cutoffs. The cubic polynomials are summed and individually scaled by the coefficients, and by virtue of its form and the heaviside step function, they cut off at the desired radius.

Listing P.6: Cubic Splines

---

```

1 #TYPE cubic_spline
2 #P -165.0 -78.5 -78.15 1.868
3 #PF 0.976 1.15 1.216 1.650

```

---

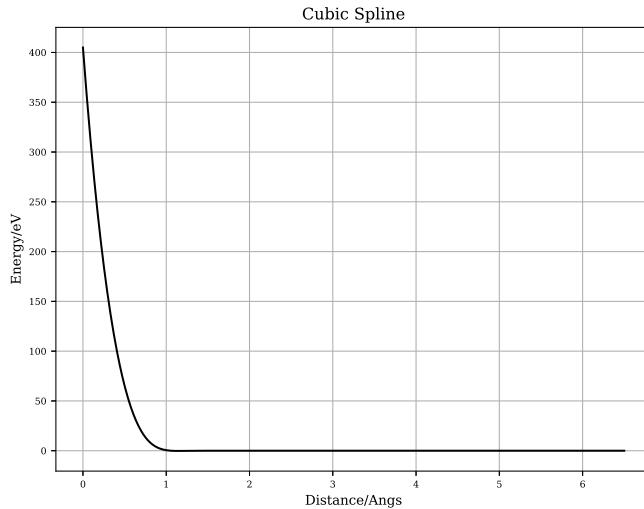


Figure P.6: Ackland Embedding

### P.2.7 Cubic Spline Plus ZBL

$$V(r) = \begin{cases} z(r) & r < r_1 \\ a_0 + a_1r + a_2r^2 + a_3r^3 & r_1 \leq r \leq r_2 \text{ where} \\ \sum_i^N a_i(r - r_i)^3 H(r_i - r) & r > r_2 \end{cases} \quad (P.7)$$

$$H(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

This function requires two sets of parameters. P is a list of N coefficients and PF is a list of N cutoffs. The cubic polynomials are summed and individually scaled by the coefficients, and by virtue of its form and the heaviside step function, they cut off at the desired radius.

Listing P.7: Cubic Splines Plus ZBL

---

```

1 #TYPE cubic_spline_zbl
2 #P 0.707937643181 1.00089008639 0.978759224081 -0.0460365116314 -0.00762125172653 -0.0209035950064
   0.00938859724754 0.000854708875904 0.0230647235533 0.011459696349 0.023419434017 0.0220932107774
   0.0325043051929 -0.0425828941427 -0.0147289840464 0.0138435254515 0.000880298224106 -0.0134658842396
   0.00102524605929
3 #PF 1.7 1.8 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 3.0 3.3 3.7 4.2 4.7 5.3 6.0 6.5 26.0 26.0 0.55 1.7 1.0

```

---

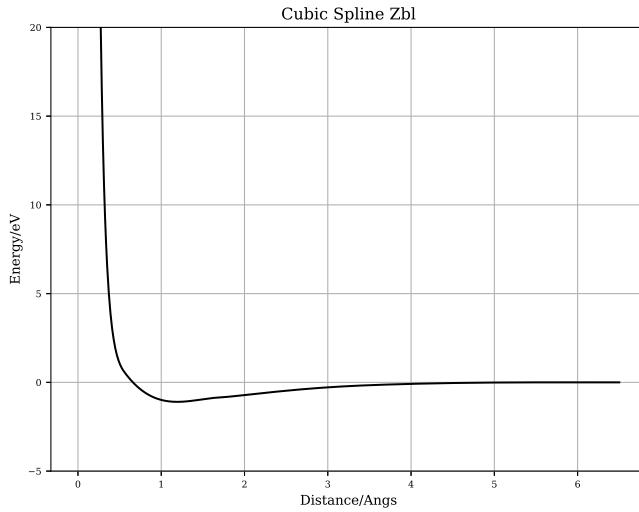


Figure P.7: Ackland Embedding

Listing P.8: Cubic Splines Plus ZBL

---

```

1 #TYPE cubic_spline_zbl
2 #P 1.00048374974 0.999953842306 0.817335736705 -0.0469506463728 -0.118042905174 0.00429308591859 0.024332744974
   0.0454260983505 0.0205392979832 0.415064378029 0.649740384042 0.0433627017609 0.047890256076 0.026097144285
   0.0106186840232 0.0138597059993 -0.0172453611171 -0.00144350314717 -0.00170969310596
3 #PF 1.7 1.8 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 3.0 3.3 3.7 4.2 4.7 5.3 6.0 6.5 26.0 26.0 0.6 1.6 1.0

```

---

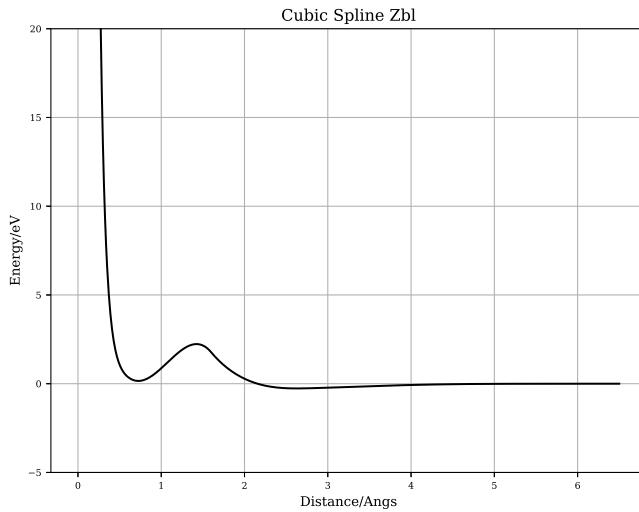


Figure P.8: Ackland Embedding

### P.2.8 Embedding A - Finnis Sinclair

$$F(\rho) = -a\sqrt{\rho} \quad (\text{P.8})$$

This function requires one parameter[110].

Listing P.9: Embedding A

---

```
1 #TYPE embedding_a
2 #P 0.1
```

---

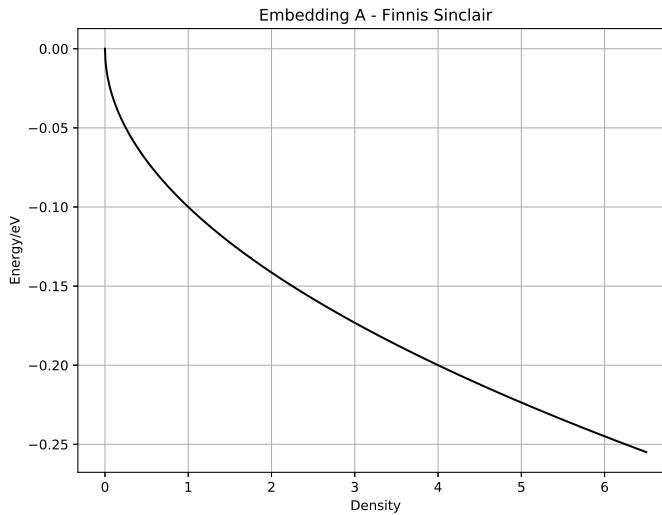


Figure P.9: Embedding A

### P.2.9 Embedding B - Mendelev

$$F(\rho) = -\sqrt{\rho} + a\rho^2 \quad (\text{P.9})$$

This function requires one parameter. Development of new interatomic potentials appropriate for crystalline and liquid iron, 2003.

Listing P.10: Embedding B

---

```
1 #TYPE embedding_b
2 #P 0.1
```

---

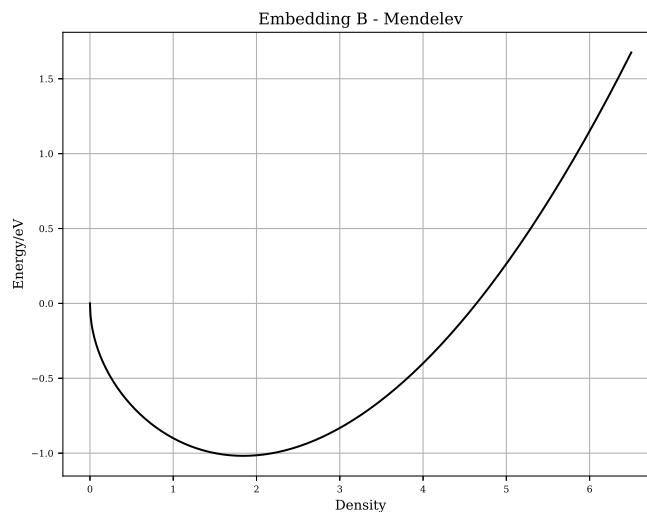


Figure P.10: Embedding B

### P.2.10 Embedding C

$$F(\rho) = a\sqrt{\rho} + b\rho^2 \quad (\text{P.10})$$

Listing P.11: Embedding C

---

```
1 #TYPE embedding_c
2 #P 0.1 0.1
```

---

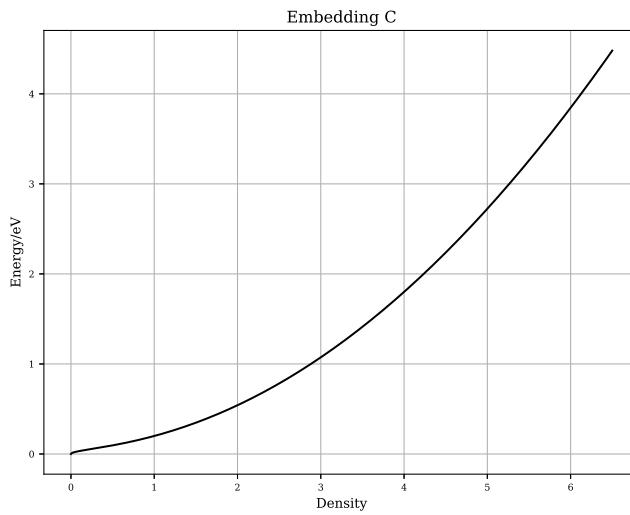


Figure P.11: Embedding C

### P.2.11 Embedding D - Ackland-Mendelev

$$F(\rho) = -\sqrt{(\rho + a\rho^2 + b\rho^4)} \quad (\text{P.11})$$

This function requires one parameter. Development of an interatomic potential for phosphorus impurities in  $\alpha$ -iron, 2004.

Listing P.12: Embedding D

---

```
1 #TYPE embedding_d
2 #P 0.05 -0.00001
```

---

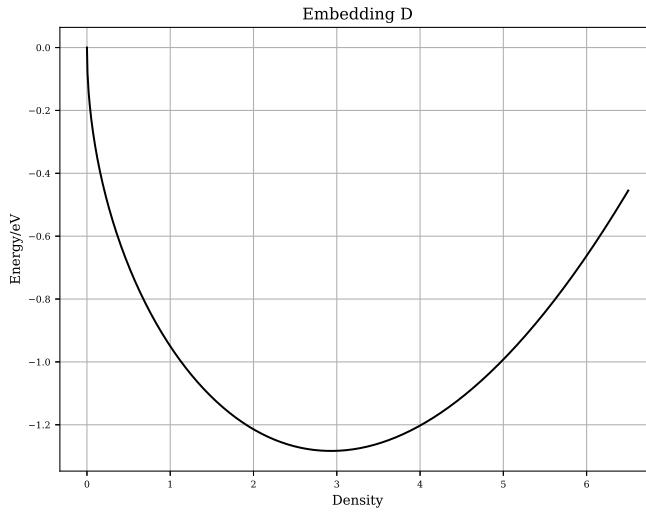


Figure P.12: Embedding D

### P.2.12 Embedding E

$$F(\rho) = a\sqrt{\rho} + b\rho^2 + c\rho^4 \quad (\text{P.12})$$

Listing P.13: Embedding E

---

```
1 #TYPE embedding_e
2 #P -0.8 0.05 -0.00001
```

---

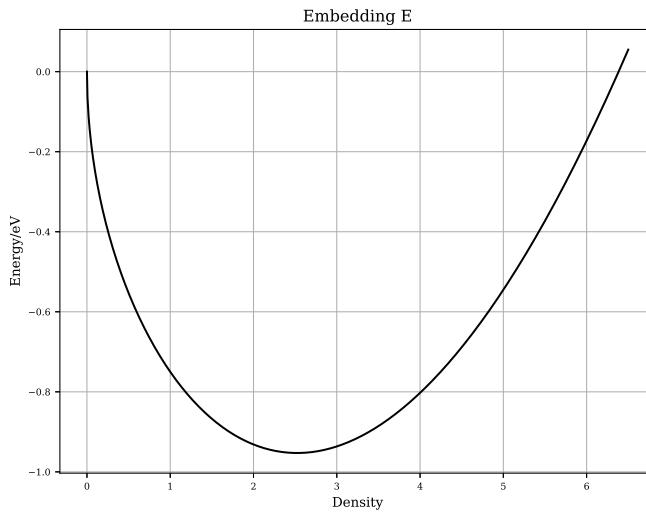


Figure P.13: Ackland Embedding

### P.2.13 Embedding F

$$F(\rho) = a\sqrt{\rho} + b\rho^2 + \rho^4 \quad (\text{P.13})$$

Listing P.14: Embedding F

---

```

1 #TYPE embedding_f
2 #P -0.8 0.05

```

---

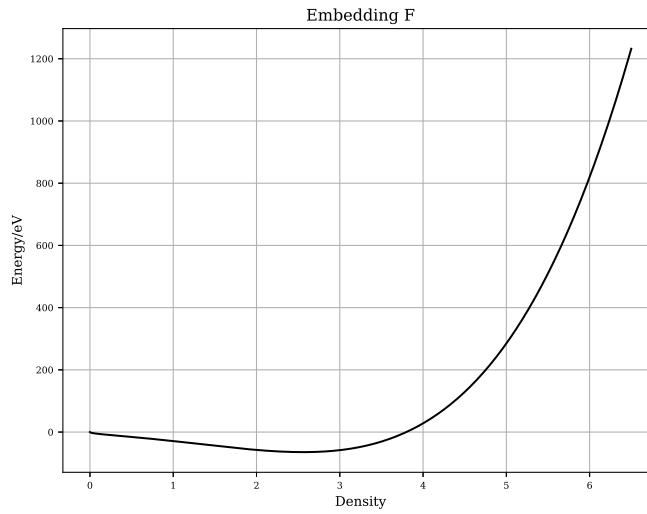


Figure P.14: Embedding F Function Plot

### P.2.14 Embedding G

$$F(\rho) = a\sqrt{\rho} + b\rho^2 + f\rho^4 \quad (\text{P.14})$$

Listing P.15: Embedding F

---

```

1 #TYPE embedding_g
2 #P -9.15 5.57
3 #PF 0.01

```

---

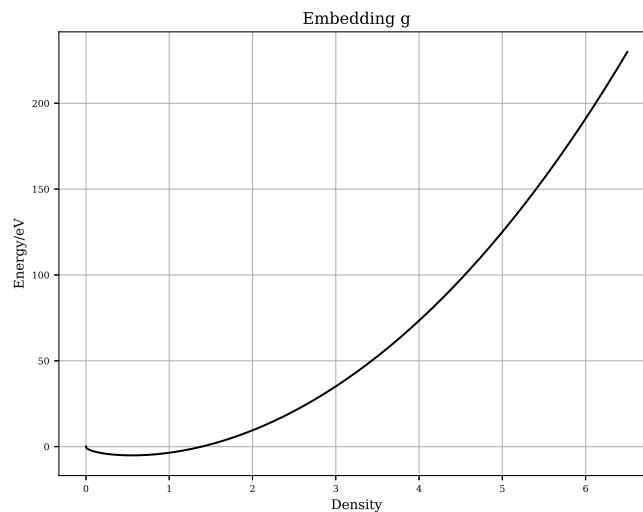


Figure P.15: Embedding G Function Plot

### P.2.15 Lennard-Jones

$$V(r) = e \left( \left( \frac{r_m}{r} \right)^{12} - 2 \left( \frac{r_m}{r} \right)^6 \right) \quad (\text{P.15})$$

Listing P.16: Lennard-Jones

---

```
1 #TYPE lennard_jones
2 #P 2.3 3.5
```

---

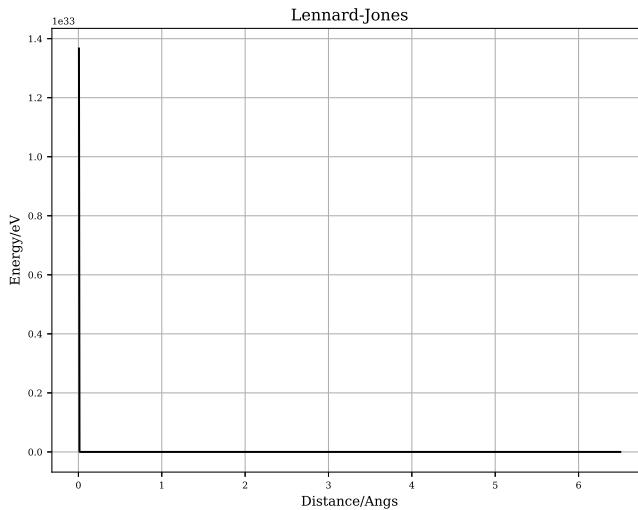


Figure P.16: Lennard-Jones

### P.2.16 Morse

$$V(r) = \exp(-2a(r - r_e)) - 2\exp(-a \times (r - r_e)) \quad (\text{P.16})$$

Listing P.17: Morse

---

```
1 #TYPE morse
2 #P 4.669 1.256 2.8
```

---

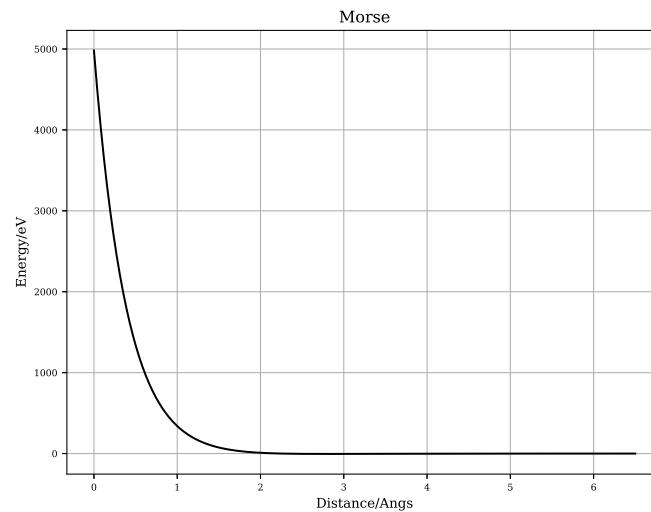


Figure P.17: Morse

### P.2.17 Slater 4S (Squared)

$$\rho(r) = (N_s r^3 \exp(-\eta r))^2 \quad (\text{P.17})$$

Listing P.18: Slater 4S

---

```
1 #TYPE slater_4s
2 #P 5.0 1.323
```

---

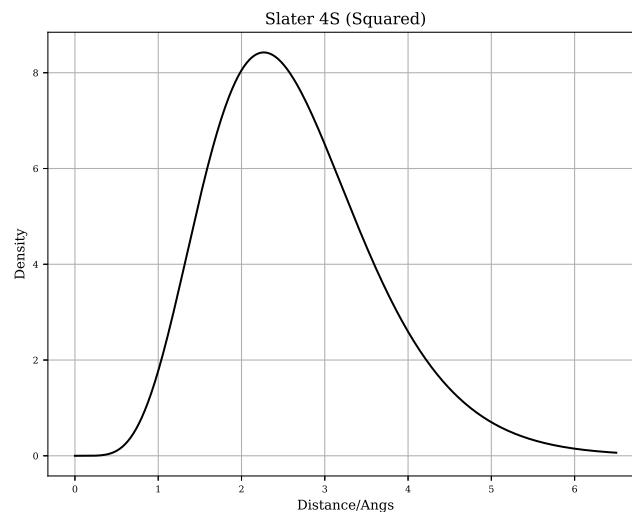


Figure P.18: Quadratic Density

### P.2.18 ZBL

$$\phi(x) = 0.181e^{-3.2x} + 0.5099e^{-0.9423x} + 0.2802e^{-0.4029x} + 0.02817e^{-0.2016x}$$

$$\text{where } a_{ij} = \frac{0.8854a_0}{Z_i^{0.23} + Z_j^{0.23}} \quad (\text{P.18})$$

and  $a_0 = 0.529$  angstrom

Listing P.19: ZBL

---

```
1 #TYPE zbl
2 #P 26.0 26.0
```

---

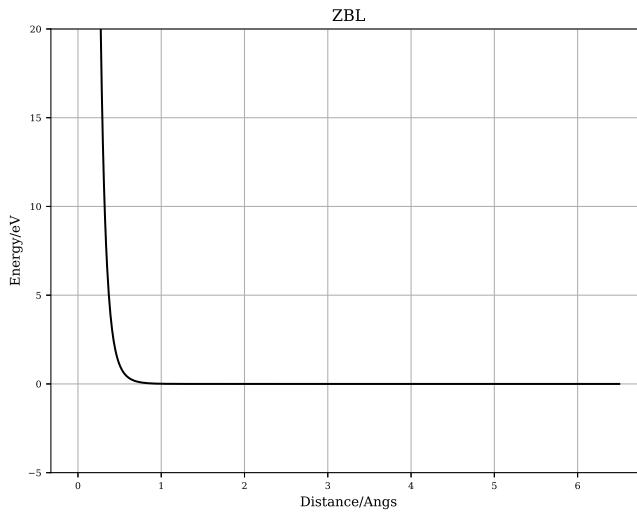


Figure P.19: ZBL

### P.2.19 Zero

$$f(x) = 0 \quad (\text{P.19})$$

Listing P.20: Zero

---

```
1 #TYPE zero
2 #P 0.0
```

---

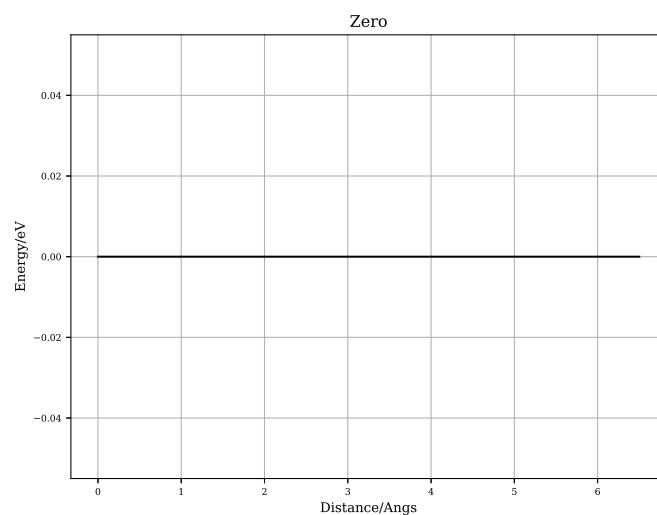


Figure P.20: Zero

# Appendix Q

## Contribution to the DL POLY Source Code

### Q.1 Contribution to DL POLY 4.04

I originally downloaded the 4.03 version of DL POLY while conducting my literature review. As a result of reading work on 2BEAM I made modifications to the Fortran code, specifically changing the arrays used to hold the potentials and retrieve the energy and forces from this modified metal potential. Dr. Todorov assisted me and adjusted my modifications such that the code worked across domains when running DL POLY over multiple processes.

Listing Q.1: Add two numbers function

---

```
1 NEW FEATURES & IMPROVEMENTS
2 -----
3
4 1. New two band (2B) EAM and EEAM potentials for metals (TEST45 and
5     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;
```

---

Listing Q.2: Add two numbers function

---

```
1 Subroutine metal_table_read(l_top)
2
3 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
4 !
5 ! dl_poly_4 subroutine for reading potential energy and force arrays
6 ! from TABEAM file (for metal EAM & EEAM forces only)
7 !
8 ! copyright - daresbury laboratory
9 ! author - w.smith march 2006
10 ! amended - i.t.todorov march 2016
11 ! contrib - r.davidchak (eeam) june 2012
12 ! contrib - b.palmer (2band) may 2013
13 !
14 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
15
16 Use kinds_f90
17 Use comms_module, Only : idnode,mxnnode,gsum
```

---

```

18 Use setup_module, Only : ntable,nrite,mxgmet,engunit
19 Use site_module, Only : ntpatm,unqatm
20 Use metal_module, Only : ntpmet,tabmet,lstmet,vmet,dmet,dmes,fmet,fmes
21 Use parse_module, Only : get_line,get_word,lower_case,word_2_real
22
23 Implicit None
24
25 Logical, Intent( In ) :: l_top
26
27 Logical :: safe
28 Character( Len = 200 ) :: record
29 Character( Len = 40 ) :: word
30 Character( Len = 4 ) :: keyword
31 Character( Len = 8 ) :: atom1,atom2
32 Integer :: fail(1:2),i,j,ipot,numpot,ktype,ngrid, &
33                      cp,cd,cds,ce,ces,katom1,katom2,keymet,k0,jtpatm
34 Real( Kind = wp ) :: start,finish
35
36 Integer,      Dimension( : ), Allocatable :: cpair, cdens,cdnss, &
37                                     cembed,cembds
38 Real( Kind = wp ), Dimension( : ), Allocatable :: buffer
39
40 fail=0
41 If      (tabmet == 1) Then ! EAM
42     Allocate (cpair(1:(ntpmet*(ntpmet+1))/2),cdens(1:ntpmet),
43                           cembed(1:ntpmet),                               &
44                           Stat=fail(1))
45 Else If (tabmet == 2) Then ! EEAM
46     Allocate (cpair(1:(ntpmet*(ntpmet+1))/2),cdens(1:ntpmet**2),
47                           cembed(1:ntpmet),                               &
48                           Stat=fail(1))
49 Else If (tabmet == 3) Then ! 2BEAM
50     Allocate (cpair(1:(ntpmet*(ntpmet+1))/2),cdens(1:ntpmet),cdnss(1:ntpmet*(ntpmet+1)/2),
51                           cembed(1:ntpmet),cembds(1:ntpmet), Stat=fail(1))
52 Else If (tabmet == 4) Then ! 2BEEAM
53     Allocate (cpair(1:(ntpmet*(ntpmet+1))/2),cdens(1:ntpmet**2),cdnss(1:ntpmet**2), &
54                           cembed(1:ntpmet),cembds(1:ntpmet), Stat=fail(1))
55 End If
56 Allocate (buffer(1:mxgmet),                               Stat=fail(2))
57 If (Any(fail > 0)) Then
58     Write(nrite,'(/,1x,a,i0)') 'metal_table_read allocation failure, node: ', idnode
59     Call error(0)
60 End If
61 cpair=0 ; cp=0
62 cdens=0 ; cd=0
63 cembed=0 ; ce=0
64 If (tabmet == 3 .or. tabmet == 4) Then
65     cdnss=0 ; cds=0
66     cembds=0 ; ces=0
67 End If
68
69 ! skip header record
70
71 Call get_line(safe,ntable,record)
72 If (.not.safe) Go To 100
73
74 ! read number of potential functions in file
75
76 Call get_line(safe,ntable,record)
77 If (.not.safe) Go To 100
78 Call get_word(record,word)

```

```

79      numpot = Nint(word_2_real(word))
80
81      Do ipot=1,numpot
82
83 ! read data type, atom labels, number of points, start and end
84
85      Call get_line(safe,ntable,record)
86      If (.not.safe) Go To 100
87
88 ! identify data type
89
90      Call get_word(record,keyword)
91      Call lower_case(keyword)
92      If (keyword == 'pair') Then
93          ktype = 1
94      Else If (keyword == 'dens' .or. keyword == 'dden') Then
95          ktype = 2
96      Else If (keyword == 'embe' .or. keyword == 'demb') Then
97          ktype = 3
98      Else If (keyword == 'sden') Then
99          ktype = 4
100     Else If (keyword == 'semb') Then
101         ktype = 5
102     Else
103         Call error(151)
104     End If
105
106 ! identify atom types
107
108     Call get_word(record,atom1)
109     If (ktype == 1 .or.                                     & ! pair
110         (ktype == 2 .and. (tabmet == 2 .or. tabmet == 4)) .or. & ! den for EEAM and dden for 2BEEAM
111         (ktype == 4 .and. (tabmet == 3 .or. tabmet == 4))) Then ! sden for 2B(EAM and EEAM)
112         Call get_word(record,atom2)
113     Else
114         atom2 = atom1
115     End If
116
117 ! data specifiers
118
119     Call get_word(record,word)
120     ngrid = Nint(word_2_real(word))
121     Call get_word(record,word)
122     start = word_2_real(word)
123     Call get_word(record,word)
124     finish = word_2_real(word)
125
126 ! check atom identities
127
128     katom1=0
129     katom2=0
130
131     Do jtpatm=1,ntpattm
132         If (atom1 == unqatm(jtpatm)) katom1=jtpatm
133         If (atom2 == unqatm(jtpatm)) katom2=jtpatm
134     End Do
135
136     If (katom1 == 0 .or. katom2 == 0) Then
137         If (idnode == 0 .and. l_top) &
138             Write(nrite,'(a)') '****',atom1,'***',atom2,'**** entry in TABEAM'
139         Call error(81)

```

```

140      End If
141
142 ! store working parameters
143
144 buffer(1)=Real(ngrid+4,wp) ! as if there are 4 extra elements after finish
145 buffer(4)=(finish-start)/Real(ngrid-1,wp)
146 buffer(2)=start-5.0_wp*buffer(4)
147 buffer(3)=finish
148
149 If (idnode == 0 .and. l_top) &
150   Write(nrite,"(1x,i10,4x,2a8,3x,2a4,2x,i6,1p,3e15.6)") &
151   ipot,atom1,atom2,'EAM-' ,keyword,ngrid,start,finish,buffer(4)
152
153 ! check array dimensions
154
155 If (ngrid+4 > mxgmet) Then
156   Call warning(270,Real(ngrid+4,wp),Real(mxgmet,wp),0.0_wp)
157   Call error(48)
158 End If
159
160 keymet=(Max(katom1,katom2)*(Max(katom1,katom2)-1))/2 + Min(katom1,katom2)
161 k0=lstmet(keymet)
162
163 ! check for undefined potential
164
165 If (k0 == 0) Call error(508)
166
167 ! read in potential arrays
168
169 Do i=1,(ngrid+3)/4
170   j=Min(4,ngrid-(i-1)*4)
171   If (idnode == 0) Then
172     Read(Unit=ntable, Fmt=*, End=100) buffer(4*i+1:4*i+j)
173   Else
174     buffer(4*i+1:4*i+j)=0.0_wp
175   End If
176 End Do
177
178 If (mxnode > 1) Call gsum(buffer(5:ngrid+4))
179
180 ! copy data to internal arrays
181
182 If (ktype == 1) Then
183
184 ! pair potential terms
185
186 ! Set indices
187
188 ! k0=lstmet(keymet)
189
190 cp=cp+1
191 If (Any(cpair(1:cp-1) == k0)) Then
192   Call error(509)
193 Else
194   cpair(cp)=k0
195 End If
196
197 vmet(1,k0,1)=buffer(1)
198 vmet(2,k0,1)=buffer(2)
199 vmet(3,k0,1)=buffer(3)
200 vmet(4,k0,1)=buffer(4)

```

```

201
202     Do i=5,mxgmet
203         If (i-4 > ngrid) Then
204             vmet(i,k0,1)=0.0_wp
205         Else
206             buffer(i)=buffer(i)*engunit
207             vmet(i,k0,1)=buffer(i)
208         End If
209     End Do
210
211 ! calculate derivative of pair potential function
212
213     Call metal_table_derivatives(k0,buffer,Size(vmet,2),vmet)
214
215 ! adapt derivatives for use in interpolation
216
217     Do i=5,ngrid+4
218         vmet(i,k0,2)=-(Real(i,wp)*buffer(4)+buffer(2))*vmet(i,k0,2)
219     End Do
220
221     Else If (ktype == 2) Then
222
223     ! density function terms
224     ! s-density density function terms for EAM & EEAM
225     ! d-density density function terms for 2B(EAM & EEAM)
226
227     ! Set indices
228
229     If      (tabmet == 1 .or. tabmet == 3) Then ! EAM
230         k0=katom1
231     Else If (tabmet == 2 .or. tabmet == 4) Then ! EEAM
232         k0=(katom1-1)*ntpatm+katom2
233     End If
234
235     cd=cd+1
236     If (Any(cdens(1:cd-1) == k0)) Then
237         Call error(510)
238     Else
239         cdens(cd)=k0
240     End If
241
242     dmet(1,k0,1)=buffer(1)
243     dmet(2,k0,1)=buffer(2)
244     dmet(3,k0,1)=buffer(3)
245     dmet(4,k0,1)=buffer(4)
246
247     Do i=5,mxgmet
248         If (i-4 > ngrid) Then
249             dmet(i,k0,1)=0.0_wp
250         Else
251             dmet(i,k0,1)=buffer(i)
252         End If
253     End Do
254
255     ! calculate derivative of density function
256
257     Call metal_table_derivatives(k0,buffer,Size(dmet,2),dmet)
258
259     ! adapt derivatives for use in interpolation
260
261     dmet(1,k0,2)=0.0_wp

```

```

262     dmet(2,k0,2)=0.0_wp
263     dmet(3,k0,2)=0.0_wp
264     dmet(4,k0,2)=0.0_wp
265
266     Do i=5,ngrid+4
267       dmet(i,k0,2)=-(Real(i,wp)*buffer(4)+buffer(2))*dmet(i,k0,2)
268     End Do
269
270   Else If (ktype == 3) Then
271
272   ! embedding function terms
273   ! s-density embedding function terms for EAM & EEAM
274   ! d-density embedding function terms for 2B(EAM & EEAM)
275
276   ! Set indices
277
278   k0=katom1
279
280   ce=ce+1
281   If (Any(cembed(1:ce-1) == k0)) Then
282     Call error(511)
283   Else
284     cembed(ce)=k0
285   End If
286
287   fmet(1,k0,1)=buffer(1)
288   fmet(2,k0,1)=buffer(2)
289   fmet(3,k0,1)=buffer(3)
290   fmet(4,k0,1)=buffer(4)
291
292   Do i=5,mxgmet
293     If (i-4 > ngrid) Then
294       fmet(i,k0,1)=0.0_wp
295     Else
296       buffer(i)=buffer(i)*engunit
297       fmet(i,k0,1)=buffer(i)
298     End If
299   End Do
300
301   ! calculate derivative of embedding function
302
303   Call metal_table_derivatives(k0,buffer,Size(fmet,2),fmet)
304
305   Else If (ktype == 4) Then
306
307   ! s-density function terms
308
309   ! The 2BM formalism for alloys allows for a mixed s-band density: rho_{atom1,atom2} /= 0
310   ! (and in general for the EEAM it may be non-symmetric: rho_{atom1,atom2} may be /= rho_{atom2,atom1})
311   ! Some 2BM models rho_{atom1,atom1}=rho_{atom2,atom2}=0 with rho_{atom1,atom2} /= 0
312   ! whereas others choose not to have mixed s-band densities.
313
314   ! Set indices
315
316   If (tabmet == 3) Then ! 2BMEAM
317     k0=lstmet(keymet)
318   Else If (tabmet == 4) Then ! 2BMEEAM
319     k0=(katom1-1)*ntpattm+katom2
320   End If
321
322   cds=cds+1

```

```

323     If (Any(cdnss(1:cds-1) == k0)) Then
324         Call error(510)
325     Else
326         cdnss(cds)=k0
327     End If
328
329     dmes(1,k0,1)=buffer(1)
330     dmes(2,k0,1)=buffer(2)
331     dmes(3,k0,1)=buffer(3)
332     dmes(4,k0,1)=buffer(4)
333
334     If (Nint(buffer(1)) > 5) Then
335
336         Do i=5,mxgmet
337             If (i-4 > ngrid) Then
338                 dmes(i,k0,1)=0.0_wp
339             Else
340                 dmes(i,k0,1)=buffer(i)
341             End If
342         End Do
343 ! calculate derivative of density function
344
345     Call metal_table_derivatives(k0,buffer,Size(dmes,2),dmes)
346
347 ! adapt derivatives for use in interpolation
348
349     dmes(1,k0,2)=0.0_wp
350     dmes(2,k0,2)=0.0_wp
351     dmes(3,k0,2)=0.0_wp
352     dmes(4,k0,2)=0.0_wp
353
354     Do i=5,ngrid+4
355         dmes(i,k0,2)=-(Real(i,wp)*buffer(4)+buffer(2))*dmes(i,k0,2)
356     End Do
357
358     End If
359
360     Else If (ktype == 5) Then
361
362 ! s-embedding function terms
363
364 ! Set index
365
366     k0=katom1
367
368     ces=ces+1
369     If (Any(cembds(1:ces-1) == k0)) Then
370         Call error(511)
371     Else
372         cembds(ces)=k0
373     End If
374
375     fmcs(1,k0,1)=buffer(1)
376     fmcs(2,k0,1)=buffer(2)
377     fmcs(3,k0,1)=buffer(3)
378     fmcs(4,k0,1)=buffer(4)
379
380     Do i=5,mxgmet
381         If (i-4 > ngrid) Then
382             fmcs(i,k0,1)=0.0_wp
383         Else

```

```

384         buffer(i)=buffer(i)*engunit
385         fmes(i,k0,1)=buffer(i)
386         End If
387     End Do
388
389 ! calculate derivative of embedding function
390
391     Call metal_table_derivatives(k0,buffer,Size(fmes,2),fmes)
392
393     End If
394
395 End Do
396
397 If (idnode == 0) Close(Unit=ntable)
398 If (idnode == 0 .and. l_top) Write(nrite,'(/,1x,a)') 'potential tables read from TABEAM file'
399
400 If      (tabmet == 1 .or. tabmet == 2) Then ! EAM & EEAM
401     Deallocate (cpair,cdens,cembed,           Stat=fail(1))
402 Else If (tabmet == 3 .or. tabmet == 4) Then ! 2B(EAM & EEAM)
403     Deallocate (cpair,cdens,cdnss,cembed,cembds, Stat=fail(1))
404 End If
405 Deallocate (buffer,                         Stat=fail(2))
406 If (Any(fail > 0)) Then
407     Write(nrite,'(/,1x,a,i0)') 'metal_table_read deallocation failure, node: ', idnode
408     Call error(0)
409 End If
410
411 Return
412
413 ! end of file error exit
414
415 100 Continue
416
417 If (idnode == 0) Close(Unit=ntable)
418 Call error(24)
419
420 End Subroutine metal_table_read

```

---

# Appendix R

## Fe-Pd Potential

This potential has been developed specifically for Fe and Pd where the atoms are in an FCC structure. Particular weighting was given to the relaxed configurations of each pure element, those with small additions of Pd to Fe and surface/slab configurations.

### R.1 Tabulated Potential

The files for the potential are large, with the LAMMPS version being over 7,000 lines long containing 35,000+ data points. They are not included here but are available to download from:

[https://github.com/BenPalmer1983/fepd\\_feru\\_potential](https://github.com/BenPalmer1983/fepd_feru_potential)

## R.2 Bulk Properties

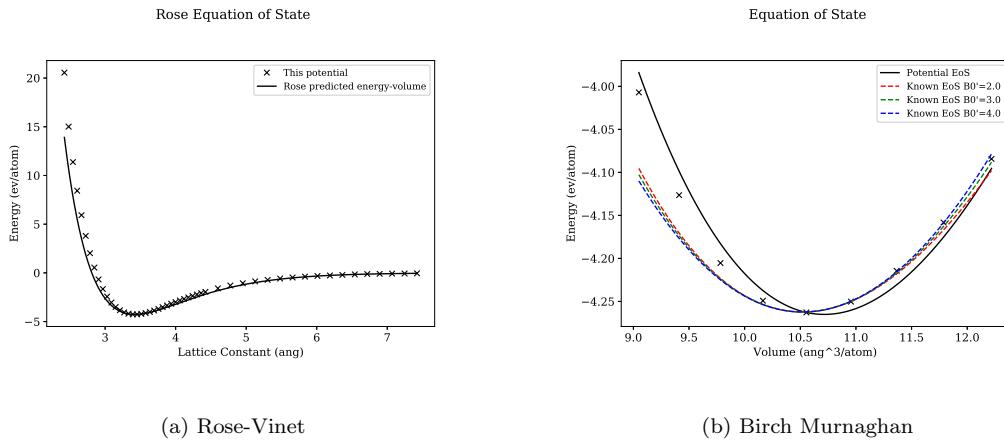
### R.2.1 Iron FCC

#### Bulk Property Values

Property	DFT Computed	This Potential
$a_0$ (angs)	3.42	3.43
Basis	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.05 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.05 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$
$E_{coh}$ (eV)	-4.26	-4.27
$B_0$ (GPa)	226.0	297.2
Stiff.1 (GPa)	$\begin{bmatrix} 364.6 & 141.6 & 233.8 & 0 & 0 & 0 \\ 141.6 & 298.7 & 130.40 & 0 & 0 & 0 \\ 233.8 & 130.4 & 364.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 186.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 266.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 186.3 \end{bmatrix}$	$\begin{bmatrix} 354.89 & 239.78 & 278.15 & 0 & 0 & 0 \\ 239.78 & 348.39 & 239.78 & 0 & 0 & 0 \\ 278.15 & 239.78 & 354.89 & 0 & 0 & 0 \\ 0 & 0 & 0 & 164.38 & 0 & 0 \\ 0 & 0 & 0 & 0 & 208.83 & 0 \\ 0 & 0 & 0 & 0 & 0 & 164.38 \end{bmatrix}$
Stiff.2 (GPa)	$\begin{bmatrix} 364.6 & 141.6 & 233.8 & 0 & 0 & 0 \\ 141.6 & 298.7 & 130.40 & 0 & 0 & 0 \\ 233.8 & 130.4 & 364.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 186.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 266.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 186.3 \end{bmatrix}$	$\begin{bmatrix} 370.46 & 260.55 & 260.55 & 0 & 0 & 0 \\ 260.55 & 370.46 & 260.55 & 0 & 0 & 0 \\ 260.55 & 260.55 & 370.46 & 0 & 0 & 0 \\ 0 & 0 & 0 & 187.87 & 0 & 0 \\ 0 & 0 & 0 & 0 & 187.87 & 0 \\ 0 & 0 & 0 & 0 & 0 & 187.87 \end{bmatrix}$

Table R.1: The lattice parameter, cohesive energy and bulk modulus were determined by fitting the Birch-Murnaghan equation of state. The elastic constants computed for the potential in stiffness (1) were computed using the method by Ravindran et al[dfttisiravindran] and in stiffness (2) by Mehl et al[100][106].

#### Equation of State Plots



(a) Rose-Vinet (b) Birch Murnaghan

Figure R.1: Equation of State plots from this potential for FCC Fe

### Elastic Constant Plots

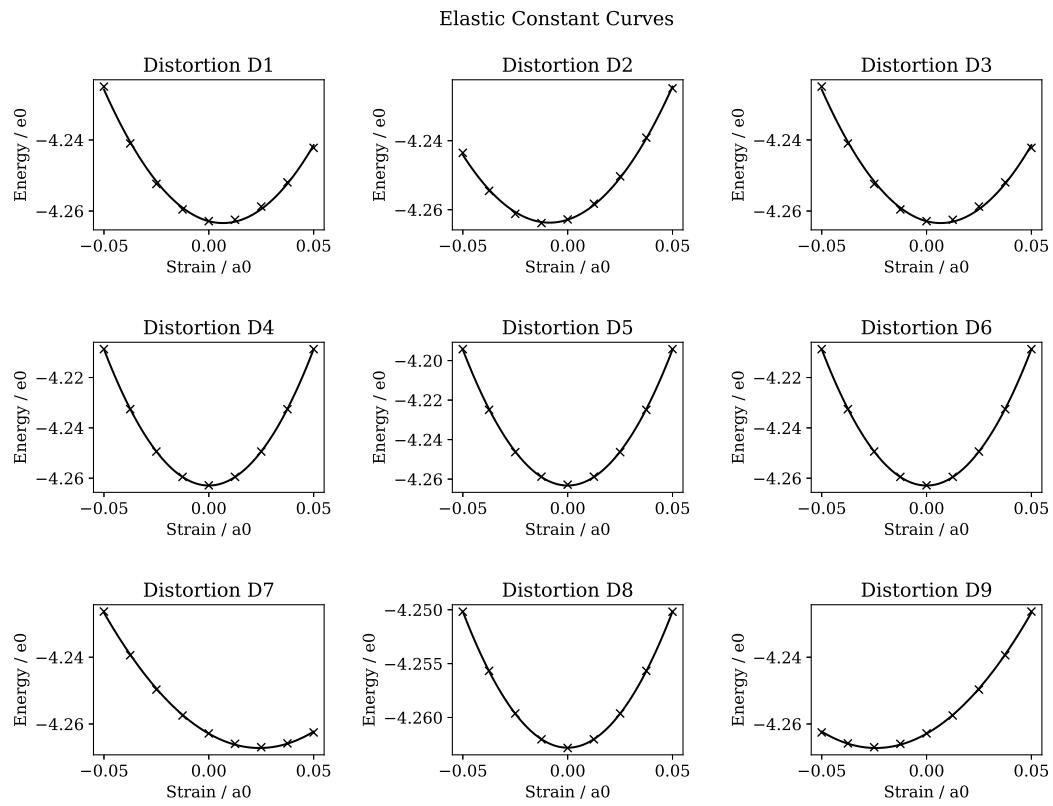


Figure R.2: Ravindran et al[dfttisiravindran] distortion-energy plots

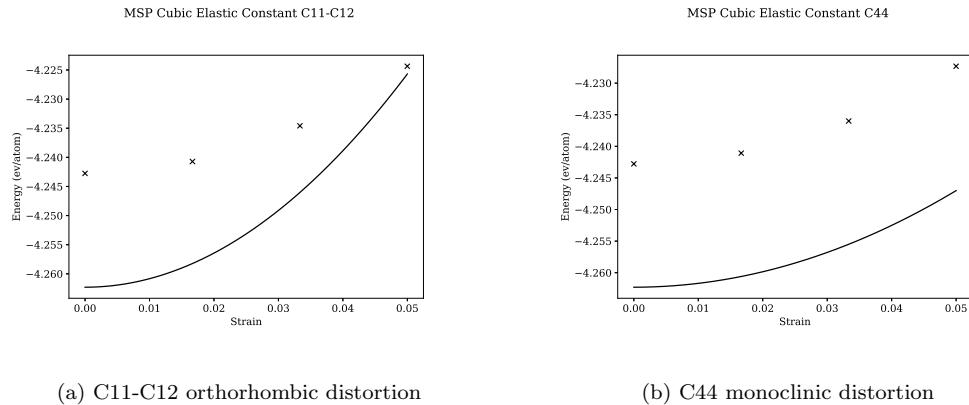


Figure R.3: Equation of State plots from this potential for FCC Fe

### R.2.2 Palladium FCC

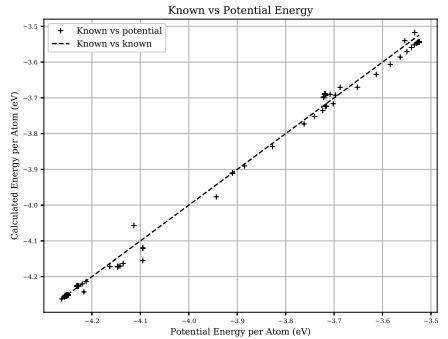
Property	DFT Computed	This Potential
$a_0$ (angs)	3.93	3.94
Basis	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$
$E_{coh}$ (eV)	-3.91	-3.92
$B_0$ (GPa)	184.0	172.3
Stiff.1 (GPa)	$\begin{bmatrix} 218.5 & 151.4 & 151.4 & 0 & 0 & 0 \\ 151.4 & 218.5 & 151.4 & 0 & 0 & 0 \\ 151.4 & 151.4 & 218.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 80.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 80.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80.3 \end{bmatrix}$	$\begin{bmatrix} 193.52 & 142.54 & 142.54 & 0 & 0 & 0 \\ 142.54 & 193.52 & 142.54 & 0 & 0 & 0 \\ 142.54 & 142.54 & 193.52 & 0 & 0 & 0 \\ 0 & 0 & 0 & 70.86 & 0 & 0 \\ 0 & 0 & 0 & 0 & 70.86 & 0 \\ 0 & 0 & 0 & 0 & 0 & 70.86 \end{bmatrix}$
Stiff.2 (GPa)	$\begin{bmatrix} 218.5 & 151.4 & 151.4 & 0 & 0 & 0 \\ 151.4 & 218.5 & 151.4 & 0 & 0 & 0 \\ 151.4 & 151.4 & 218.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 80.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 80.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80.3 \end{bmatrix}$	$\begin{bmatrix} 206.76 & 155.09 & 155.09 & 0 & 0 & 0 \\ 155.09 & 206.76 & 155.09 & 0 & 0 & 0 \\ 155.09 & 155.09 & 206.76 & 0 & 0 & 0 \\ 0 & 0 & 0 & 71.77 & 0 & 0 \\ 0 & 0 & 0 & 0 & 71.77 & 0 \\ 0 & 0 & 0 & 0 & 0 & 71.77 \end{bmatrix}$

Table R.2: The lattice parameter, cohesive energy and bulk modulus were determined by fitting the Birch-Murnaghan equation of state. The elastic constants computed for the potential in stiffness (1) were computed using the method by Ravindran et al [[dfttisiravindran](#)] and in stiffness (2) by Mehl et al [100][106].

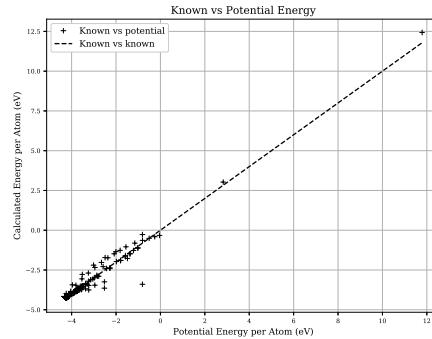
### R.2.3 Iron BCC

## R.3 Energy-Force-Stress Matching

### R.3.1 Comparing DFT energies to Potential Energies



(a) C11-C12 orthorhombic distortion



(b) C44 monoclinic distortion

Figure R.4: Equation of State plots from this potential for FCC Fe

### R.3.2 Surface Energy

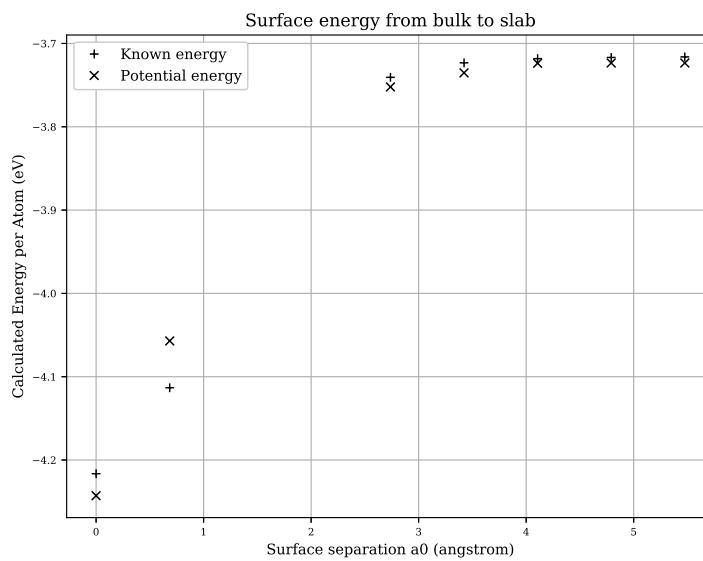
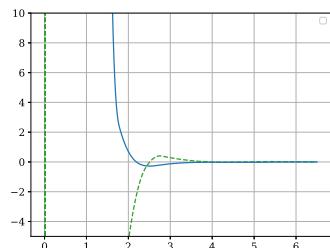
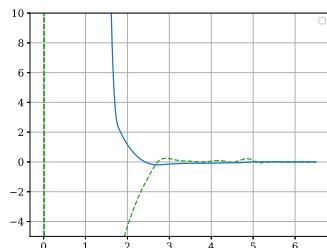


Figure R.5: Comparing the DFT energies to those calculated with this potential.

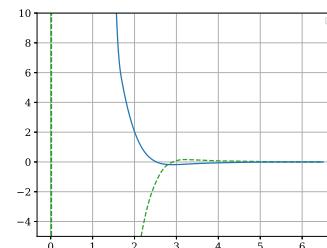
## R.4 Potential Function Plots



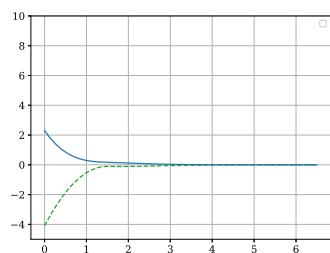
(a) Fe-Fe Pair



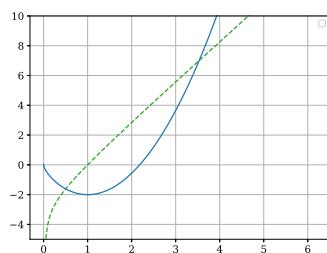
(b) Fe-Pd Pair



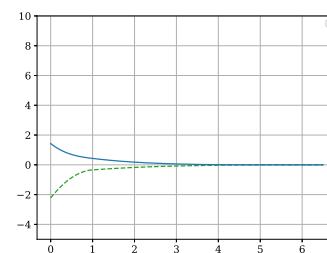
(c) Pd-Pd Pair



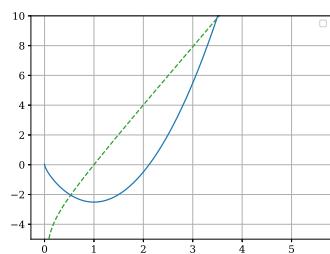
(d) Fe Density



(e) Fe Embedded



(f) Pd Density



(g) Pd Embedded

Figure R.6: Equation of State plots from this potential for FCC Fe

# Appendix S

## Fe-Pd Potential

This potential has been developed specifically for Fe and Pd where the atoms are in an FCC structure. Particular weighting was given to the relaxed configurations of each pure element, those with small additions of Pd to Fe and surface/slab configurations.

### S.1 Tabulated Potential

The files for the potential are large, with the LAMMPS version being over 7,000 lines long containing 35,000+ data points. They are not included here but are available to download from:

[https://github.com/BenPalmer1983/fepd\\_feru\\_potential](https://github.com/BenPalmer1983/fepd_feru_potential)

### S.2 Bulk Properties

#### S.2.1 Iron FCC

Property	DFT Computed	This Potential
$a_0$ (angstrom)	3.42	3.43
Basis vectors	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.05 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.05 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$
$E_{coh}$ (eV)	-4.26	-4.27
$B_0$ (GPa)	226.0	297.2
Stiffness (1) (GPa)	$\begin{bmatrix} 364.6 & 141.6 & 233.8 & 0 & 0 & 0 \\ 141.6 & 298.7 & 130.40 & 0 & 0 & 0 \\ 233.8 & 130.4 & 364.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 186.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 266.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 186.3 \end{bmatrix}$	$\begin{bmatrix} 364.6 & 141.6 & 233.8 & 0 & 0 & 0 \\ 141.6 & 298.7 & 130.40 & 0 & 0 & 0 \\ 233.8 & 130.4 & 364.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 186.3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 266.8 \\ 0 & 0 & 0 & 0 & 0 & 186.3 \end{bmatrix}$
Stiffness (2) (GPa)	$\begin{bmatrix} 364.6 & 141.6 & 233.8 & 0 & 0 & 0 \\ 141.6 & 298.7 & 130.40 & 0 & 0 & 0 \\ 233.8 & 130.4 & 364.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 186.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 266.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 186.3 \end{bmatrix}$	$\begin{bmatrix} 364.6 & 141.6 & 233.8 & 0 & 0 & 0 \\ 141.6 & 298.7 & 130.40 & 0 & 0 & 0 \\ 233.8 & 130.4 & 364.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 186.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 266.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 186.3 \end{bmatrix}$

Table S.1: The lattice parameter, cohesive energy and bulk modulus were determined by fitting the Birch-Murnaghan equation of state. The elastic constants computed for the potential in stiffness (1) were computed using the method by Ravindran et al[dfttisiravindran] and in stiffness (2) by Mehl et al[100][106].