UNIVERSITY OF BIRMINGHAM

Department of Metallurgy & Materials

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

by Ben Palmer

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

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

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

by

Ben Palmer

Abstract

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

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

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

Contents

List of Figures

List of Tables

Chapter 1

Introduction

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

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

Nuclear Power in the UK

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

Generation 1 Reactors

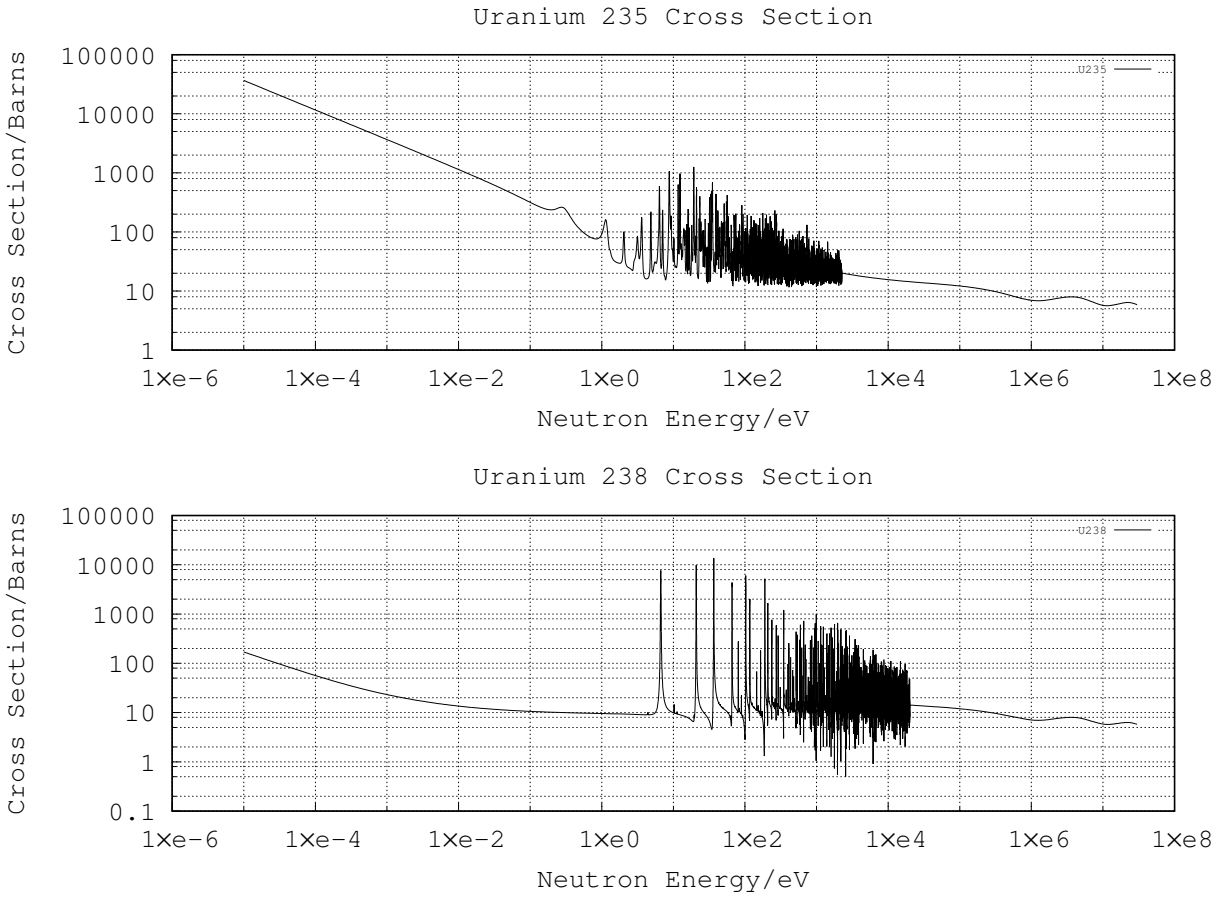


Figure 1.1: Graph caption

Generation 2 Reactors

Generation 3 Reactors

An Approaching Energy Gap for the UK

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

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

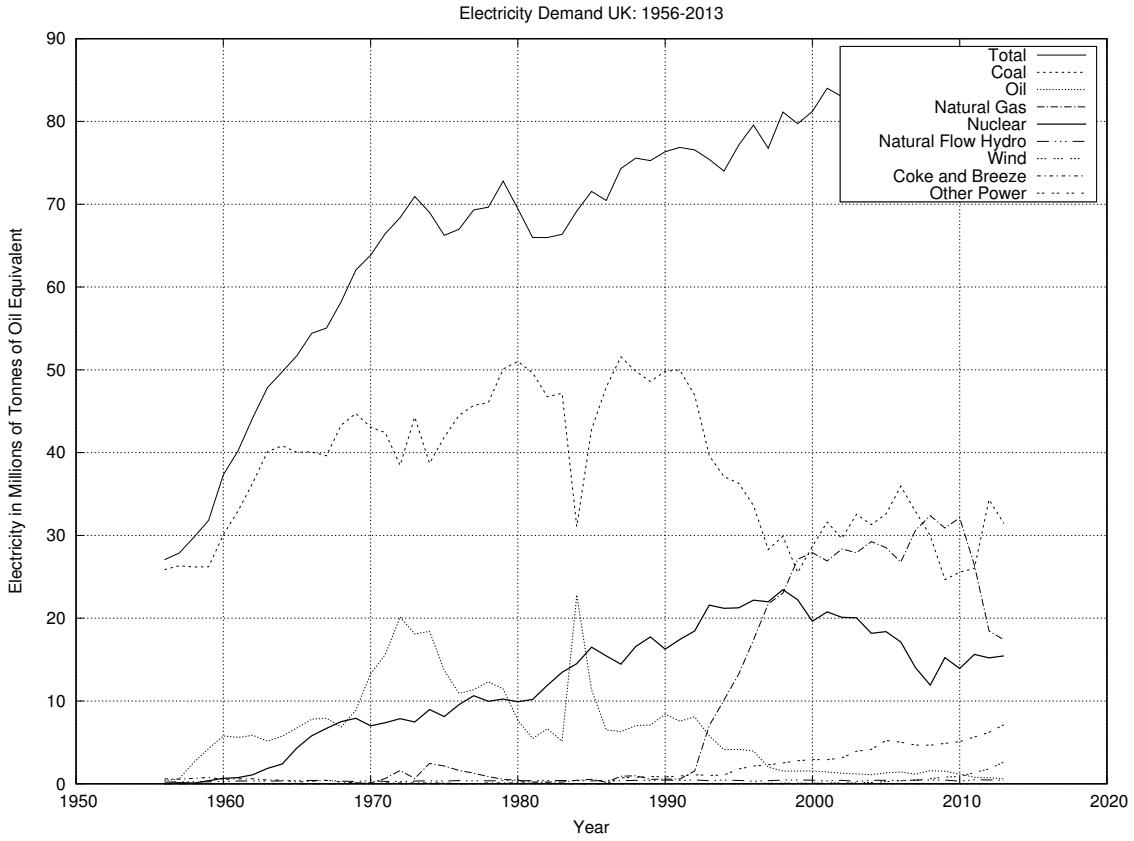


Figure 1.2: Electricity in Millions of Tonnes of Oil Equivalent

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

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

Proposed Generation III+ Nuclear Power Plants

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

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

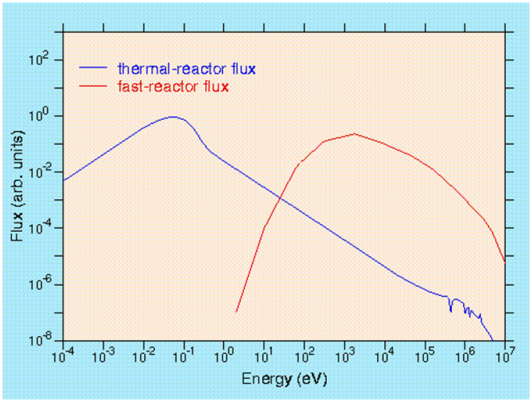


Figure 1.3: Graph caption

EPR

AP1000

Generation IV Proposed Designs

Super-Critical Water Reactors (SCWR)

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

PWR 288-325oC 15.2 MPa[2]

BWR 278-287oC 7.1MPa[3]

Lead-cooled Fast Reactors

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

Gas Cooled Fast Reactors

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

Experimental Fusion Reactors

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

$$ {}_1D^2 +_1 T^3 \rightarrow_2 He^4(3.5MeV) +_0 n^2(14.1MeV) \tag{1.1} $$

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

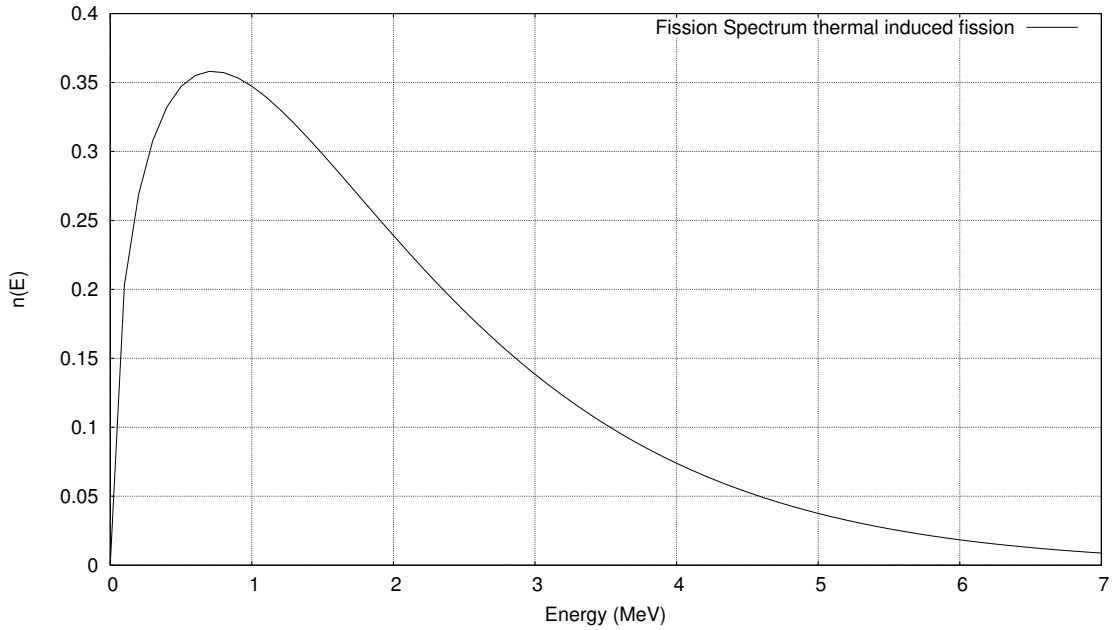


Figure 1.4: Fission Spectra

Nuclear Fission and Neutron Spectra

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

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

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

reference literature neutron damage

Neutrons

Materials Developed for Nuclear Power

With temperatures of several hundred degrees or more, pressures exceeding 150 atomspheres and a high levels of irradiation, $1.0x10^{1}1$ neutrons per "cm"^{2} per second, the interior of a nuclear reactor is a hostile and damaging environment. The choice of material is further complicated due to a number of factors.

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

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

Austenitic Stainless Steel and Reactors

Stainless Steels

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

Passive Film Protection of Chromium

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

Ferritic Stainless Steel

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

Martensitic Stainless Steel

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

Duplex Stainless Steel

Austenitic Stainless Steel

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

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

Austenitic Stainless Steels in the Nuclear Industry

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

[**eprreactoroverview**]

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

Low cobalt content

Issues Associated with Austenitic Stainless Steels

Chromium Sensitization

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

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

Irradiation Damage inside the Reactor Core

Doping with Palladium, Ruthenium and other Platinum Group Metals

Sensitization of Austenitic Stainless Steels

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

Palladium and Ruthenium at the Grain Boundary

Doping Alloys with Palladium and Ruthenium

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

Cathodic Modification

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

Effect on Intergranular Stress Corrosion Cracking

Other Notable Corrosion Resistance Enhancing Alloys

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

Ion Irradiation

Emulating Neutron Radiation

reference literature neutron damage

Neutrons

Proton Activation

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

Evaluated Nuclear Data Files

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

PADF

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

TALYS

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

TENDL

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

Scope and Objectives

Chapter 2

Background: Radiation Effects and Transport

Chapter Summary

Radiation Types Relevant to This Work

Introduction

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

Protons and Ions

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

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

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

Neutrons

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

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

Table 2.1: Neutron Categories by Energy Range [5]

Electrons

Gammas

Chapter 3

Background: Proton Activation and Radioactive Decay

Chapter Summary

Proton Accelerators

Linac

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

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

Cyclotron

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

Synchrotron

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

Radio Frequency Quadrupole

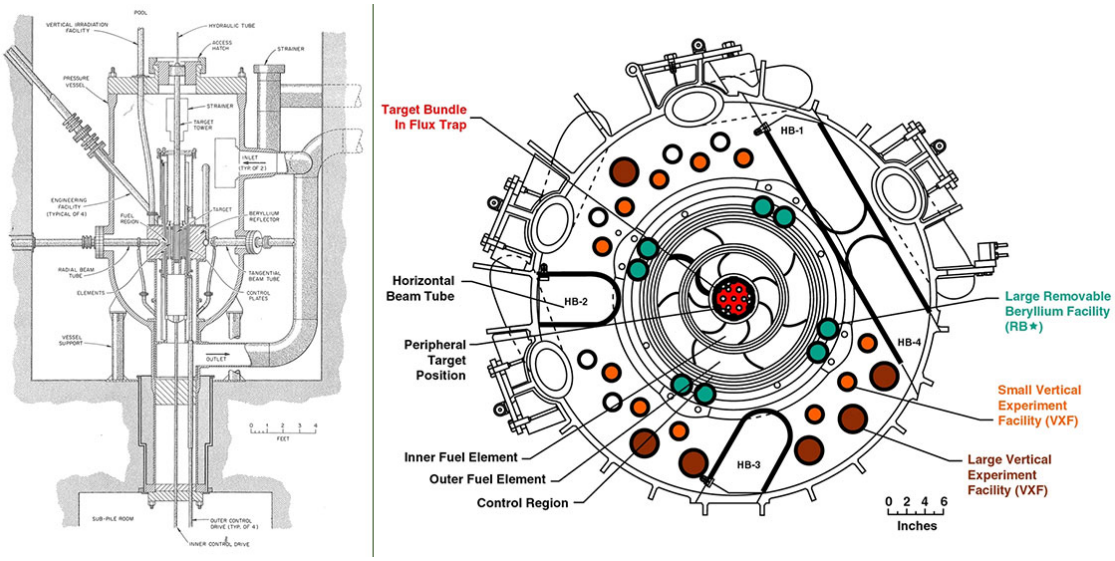


Figure 3.1: A cross section of the HFIR [8]

Neutron Sources

High-Flux Neutron Reactors

There are 250 or so research reactors in 55 countries [6], 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[7]. In Cadrache, 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 [6].

High Flux Isotope Reactor, Oak Ridge

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

NIST Center for Neutron Research

The NBS (National Bureau of Standards) Reactor was designed as a research reactor with a power output of 40MW and a neutron flux of approximately $1.0x10^15$ neutrons $cm^{-2}s^{-1}$. In 2006 it was decided to extend the facility and add beam lines

Spallation

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

ISIS

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

Source Review

Table

Table 3.1: Nbnd settings

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

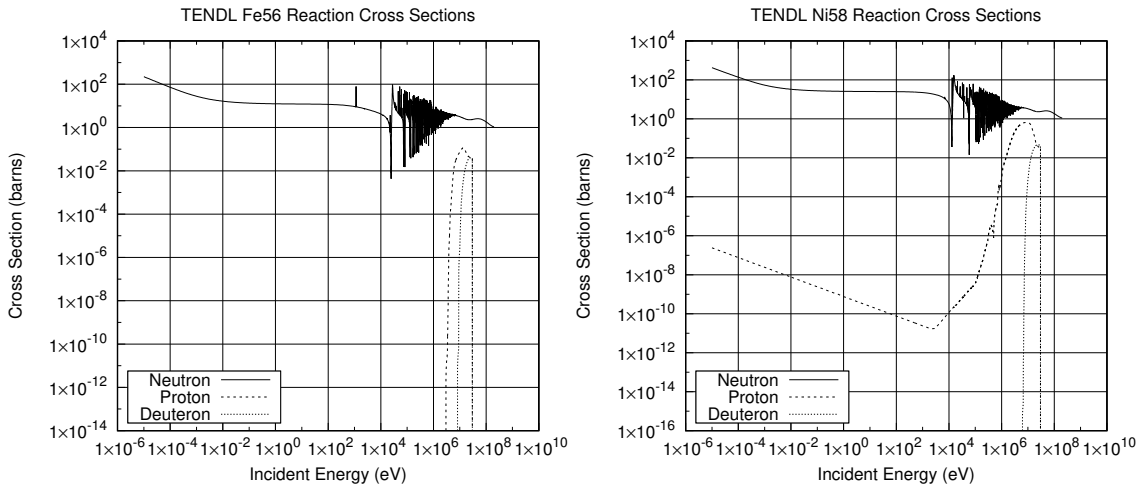


Figure 3.2: Electricity in Millions of Tonnes of Oil Equivalent

Ion Irradiation to Investigate Neutron Damage

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

Ion Irradiation at the University of Birmingham

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

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

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

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

Transmutation of Nuclei by Neutrons and Protons

Neutron Activation

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

Proton Activation

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

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

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

Radioactive Decay

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

$$\lambda = -\frac{N'(t)}{N(t)} \tag{3.2}$$

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:

$$N(t) = N(0) \exp(-t\lambda) \tag{3.3}$$

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:

$$A(t) = -N'(t) = \lambda N(t) \tag{3.4}$$

$$A(t) = \lambda N(0) \exp(-t\lambda) \tag{3.5}$$

Bateman Equation for Radioactive Decay

The English mathematician Harry Bateman derived an equation (3.6) to calculate the amount of each isotope in a decay chain, illustrated in Figure 3.3, at time t.

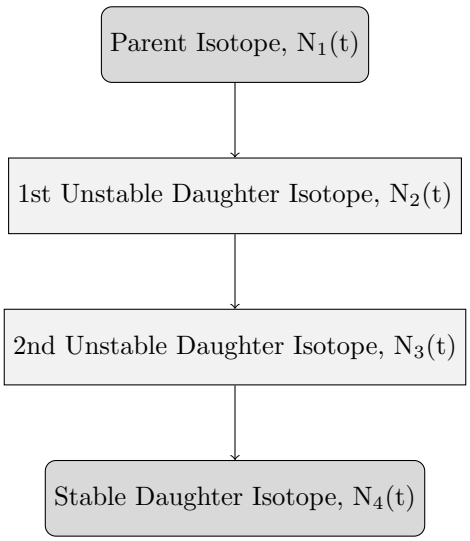


Figure 3.3: An example decay chain from an unstable parent isotope, through unstable daughter isotopes ending with a stable daughter isotope.

$$N_n(t) = \sum_{i=1}^{i=n} \left(\left(\prod_{j=i}^{j=n-1} \lambda_{(ij+1)} \right) \sum_{j=i}^{j=n} \left(\frac{N_{i0} \exp(-\lambda_j t)}{\prod_{p=i,p\neq j}^{p=n} (\lambda_p - \lambda_j)} \right) \right) \tag{3.6}$$

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) [9].

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.

Simulating Ion Irradiation with SRIM

Move to Method

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

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

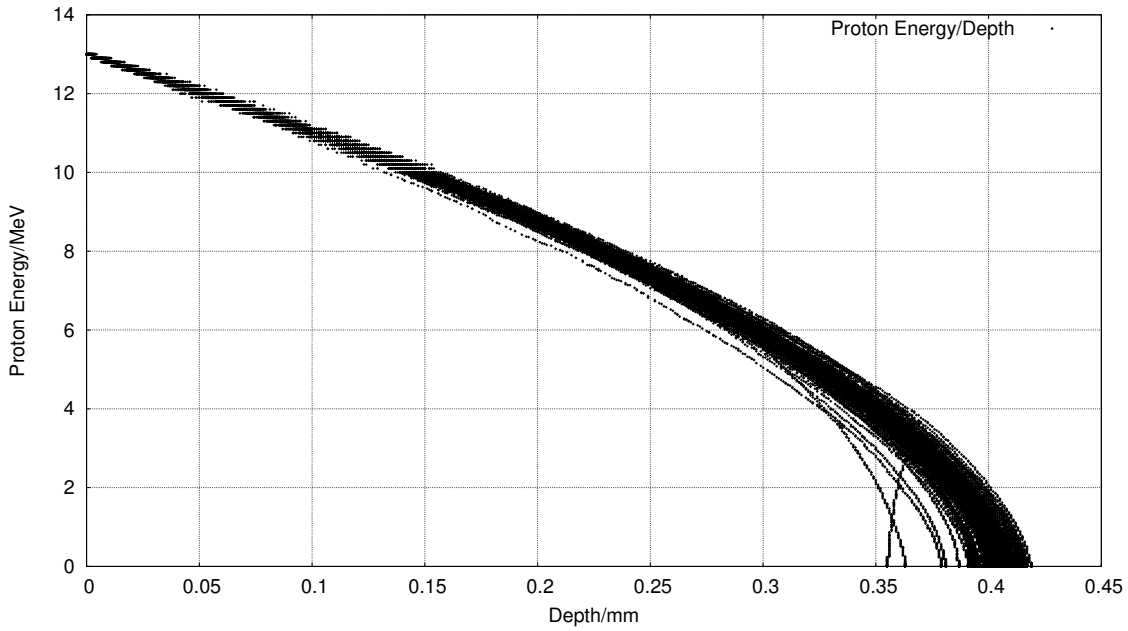


Figure 3.4: One hundred simulated 13MeV proton energy loss curves in Fe simulated with SRIM [10]

Chapter 4

Background: Interatomic Potential Fitting

Chapter Summary

Experiment, Modelling and Theory

Introduction

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

Design a size and time scale diagram

Simulating Materials on a Variety of Scales in Time and Space

Force Matching

To derive a potential, one may approach the problem from first principles in an attempt to replicate reality. It has been more useful, however, to lose any physical elegance [11] to give potentials that work for specific elements under certain conditions. Force data, gathered experimentally or by first-principles calculations, has been used to develop potentials since the 1990s. The force matching method was developed in 1994 by Ercolessi and Adams [12] to link the more accurate, more processor and memory intensive, world of first-principles calculations to Molecular Dynamics.

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

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

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

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

The overall match is measured by adding Z_F and Z_C

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

The additional contraints used in this method are typically the bulk modulus[12], the three elastic constants (C_11, C_12 and C_44) for cubic crystals [12] [13], stress tensors and phonons [13].

Crystal Properties

Introduction

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

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

Bulk Modulus

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

$$B_0 = -V\frac{\partial P}{\partial V} \tag{4.4}$$

$$B_0 = V\frac{\partial^2 E}{\partial V^2} \tag{4.5}$$

Table 4.1: Useful Conversion Factors

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

Equation of State

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

There have been various

Murnaghan Equation of State

Hooke's law implies a linear relationship between stress and strain. In practice, where a pressure is applied to a material, the application of Hooke's law is limited [14]. To improve upon formulae developed in the 1930's, Muraghan derived a new equation (expand).

Birch-Murnaghan

[15]

Choice of Equation

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

Strain

Strain definition.

Stress

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

Voigt Notation

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

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

Elastic Constants

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

$$\vec{\sigma_{ij}} = \vec{C_{ijkl}} \vec{\epsilon kl} \tag{4.7}$$

elastic constants tensor.

Magnetism

Brief History of Magnetism

Magnetic minerals have been known to and used by civilzations for millenia. Magnetite (lodestone) is a naturally occuring iron ineral.

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

Origin of Magnetism

Bohr Magneton

The Bohr magneton is a unit for measuring magnetic moment.

1 Bohr magneton = $5.79eV/T$

Electron Motion

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

$$\oint \vec{B} \cdot dl = \mu_0 I \tag{4.8}$$

Intrinsic Magnetic Moment

Ferromagnetism and Antiferromagnetism

Hund's Rule

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

Density Functional Theory

Density Functional Theory (DFT) is a branch of quantum chemistry that approximately solves the Schrdinger equation using electron density, rather than the coordinates of each electron in the system. There are many other simplifications that must be added 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.

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 Hohenburg proved that the potential of a system is uniquely determined by its ground state density. This makes solving the Schrdinger equation significantly easier.

Time Independent Schrdinger Equation

The Schrdinger equation of a system contains all the information about that system. For all but the simplest systems, it is a near impossible task to solve the Schrdinger equation.

Hohenberg-Kohn Theorem

Born-Oppenheimer

pseudopotentials

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

Reduces the number of electrons. Simplifies the potential.

DFT: Magnetism

Collinear

Non-Collinear

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

Classical Molecular Dynamics

Verlet Timestep

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

  ...

Kinetic Monte Carlo

Interatomic Potentials

Introduction

Classical molecular dynamics simulations are used to study small volumes of a material, with model sizes typically in the range of 10^5 to 10^6 atoms and time periods on the picosecond scale [16]. Such simulations have been used to model grain boundaries, and within these models irradiation damage may be simulated by initiating atom cascades that result from neutron radiation. Trautt and Mishin used molecular dynamics to study grain boundary migration in copper (12), and a more relevant example is the study by Shibuta et al and their model of grain boundary energy in bcc iron-chromium (13).

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

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

Morse and Lennard Jones Pair Potentials

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

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

Finnis Sinclair Potentials

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

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

Embedded Atom Method

$$\begin{aligned} U_{EAM} &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N} V_{ij}(r_{ij}) + \sum_{i=1}^{N} F[\rho_i] \\ &\text{where } \rho_i = \sum_{j=i,j\neq i}^{N} \rho_{ij}(r_{ij}) \end{aligned} \tag{4.9}$$

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

Two Band Embedded Atom Method

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

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

$$\begin{aligned} U_{EAM} &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N} V_{ij}(r_{ij}) + \sum_{i=1}^{N} F_D[\rho_{d,i}] + \sum_{i=1}^{N} F_S[\rho_{s,i}] \\ &\text{where } \rho_{d,i} = \sum_{j=i,j\neq i}^{N} \rho_{d,ij}(r_{ij}) \text{ and } \rho_{s,i} = \sum_{j=i,j\neq i}^{N} \rho_{s,ij}(r_{ij}) \end{aligned} \tag{4.10}$$

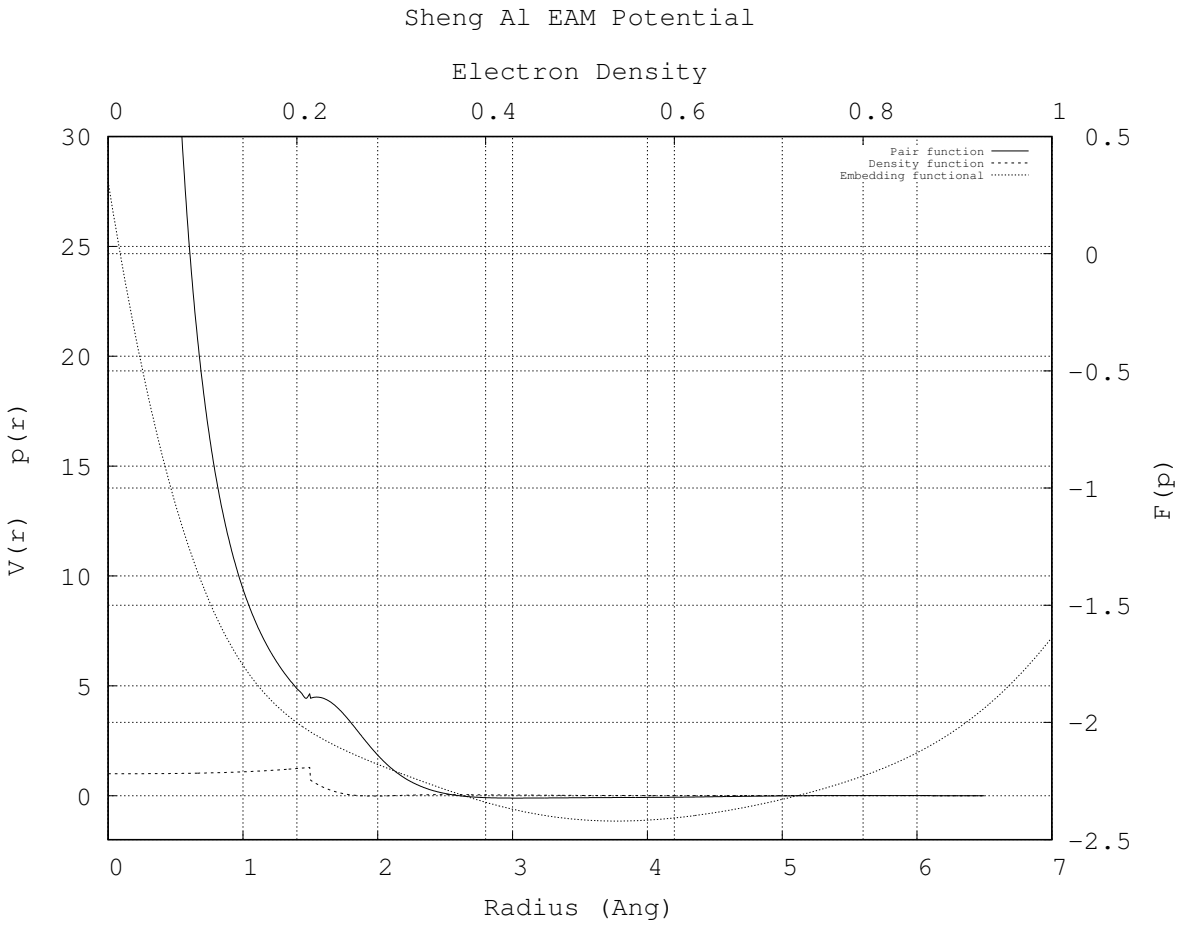


Figure 4.1: Graph caption

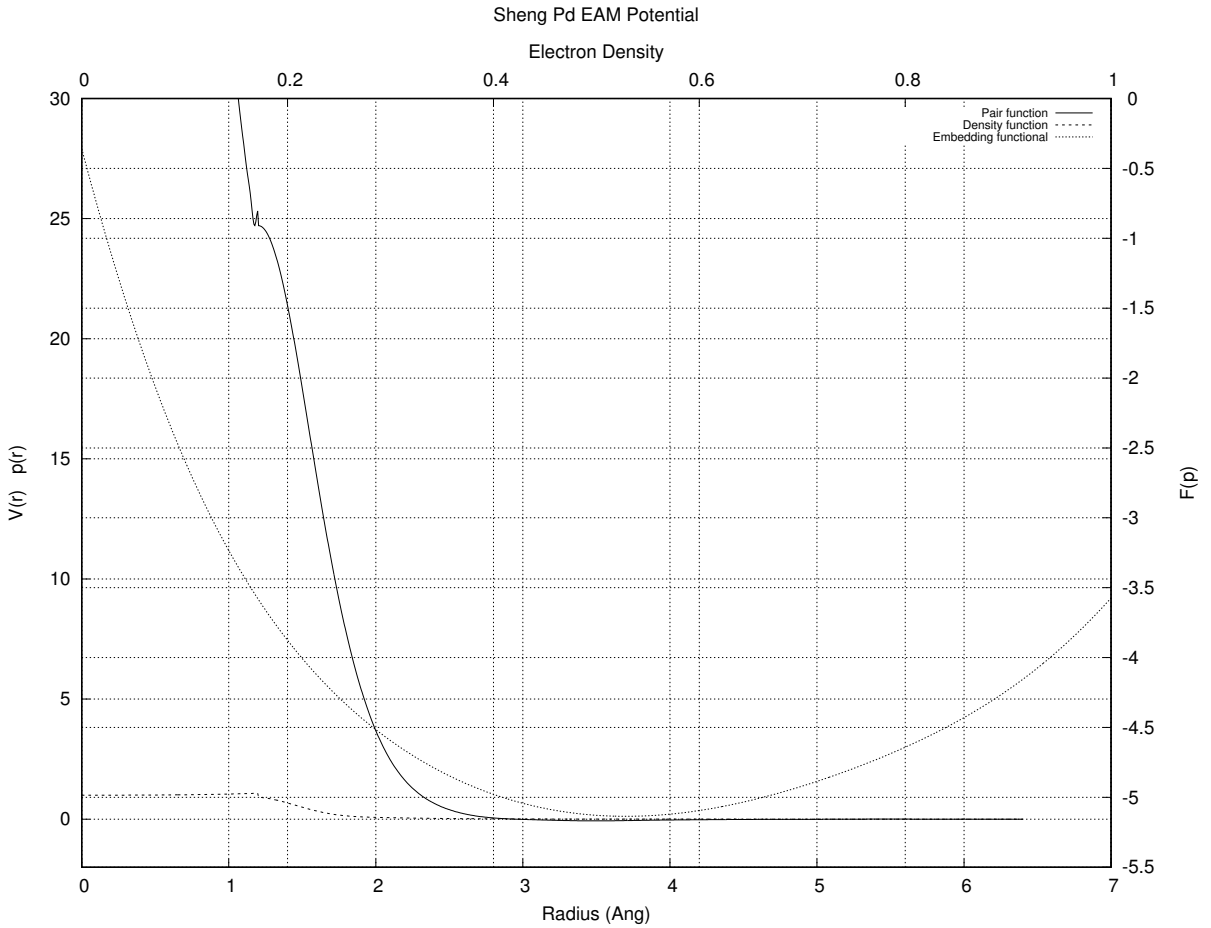


Figure 4.2: Graph caption

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

ZBL Function

It became clear, while experimenting with a number of existing potentials and molecular dynamics programs, that at the very least modifications to those potentials would need to be made for small atom seperations. 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 seperation was out of the range of the potential.

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

Functional Forms of Embedded Atom Method Potentials

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

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

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

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

$$f(r) = \sum_{i=1}^{n} A_i(r_i - r)H(r_i - r) \tag{4.11}$$

$$f(r) = \sum_{i=1}^{n} (a_i r^3 + b_i r^2 + c_i r + d_i)H(r_i^{upper} - r)H(r - r_i^{lower}) \tag{4.12}$$

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

$$f(r) = \sum_{i=1}^{n} (a_i r^3 + b_i r^2 + c_i r + d_i)H(r_i^{upper} - r)H(r - r_i^{lower}) \tag{4.13}$$

Fitting Interatomic Potentials

Analysing a Potential

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

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

Continuous Optimization

Introduction

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

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

Equations here

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

Continuous and Discrete Optimization

The difference is perhaps best described by giving two counter examples.

The traveling salesman is a well known optimisation problem where the aim is to calculate the shortest route between a set of cities. This is a discrete optimization problem, as there is a fixed and finite set of solutions.

Global and Local Optimization

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

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

Line Search

Gradient Descent

Newton Gauss

The Newton Gauss method is typically used to find a local optimum, and minimises the residual square sum between the function values and the calculated values using the parameters being optimised. It requires the calculation of the Jacobian which is then in turn used to approximate the Hessian, which would otherwise be a compute intensive process.

Levenberg Marquardt

Genetic Algorithm

The central idea behind the theory of evolution is that all life has a common ancestry, but over billions of years through natural selection,

Chapter 5

Methodology: Proton Activation and Radioactive Decay

Chapter Summary

Introduction

summary of the aim of this section

Fortran

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

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

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

Activation by Ion Irradiation

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

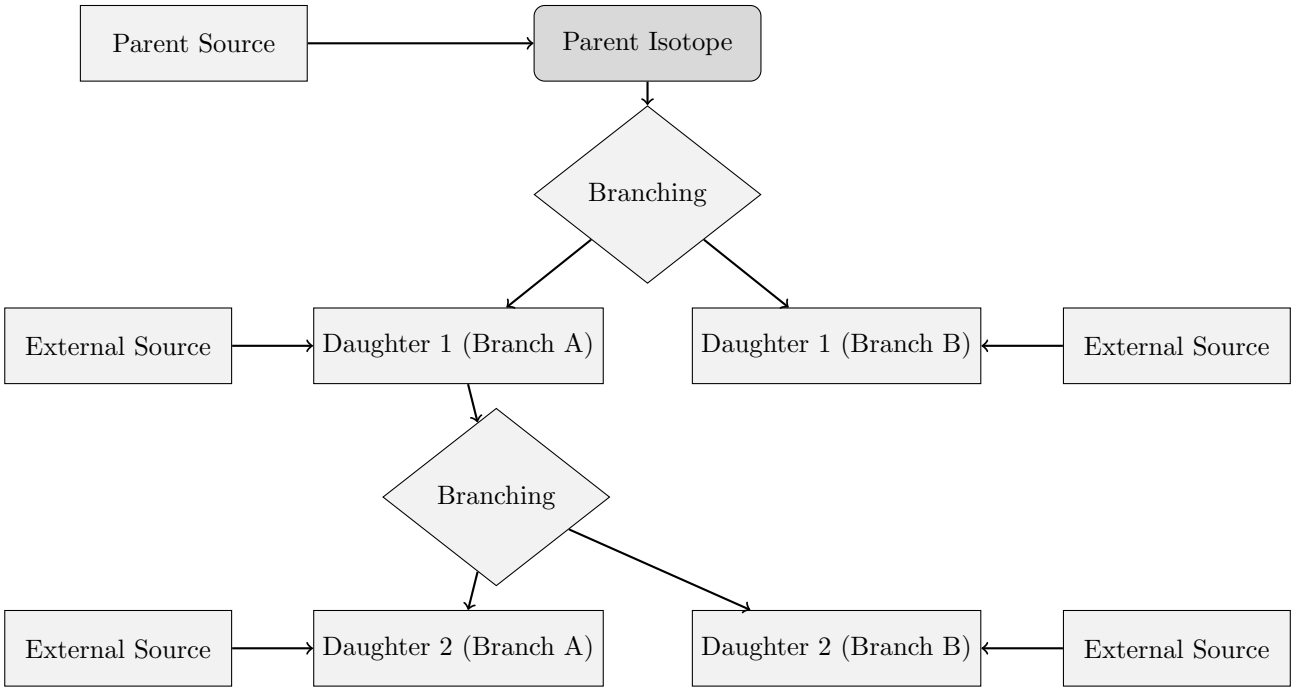


Figure 5.1: An example of several decay chains including branching factors and possible external source terms for each isotope on each chain.

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.

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

Constructing the Differential Equations

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

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

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

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

$$N_1'(t) = \omega_1 - \lambda_1 N_1(t) \tag{5.2}$$

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

$$N_z'(t) = \omega_z + b_{z-1}\lambda_{z-1}N_{z-1}(t) \tag{5.4}$$

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

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

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

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

Numerical Inversion of the Laplace Transform

The Gaver-Stehfest[17] 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).

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

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

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

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 daugher and stable daughter isotopes).

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):

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

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

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

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

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

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

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

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

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

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

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

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

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

Preference: Analytic over Numeric

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

Figure 5.2: Decay of Po-218: Analytice and Gaver-Stehfest Calculations [9]

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

Computational Methods

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

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

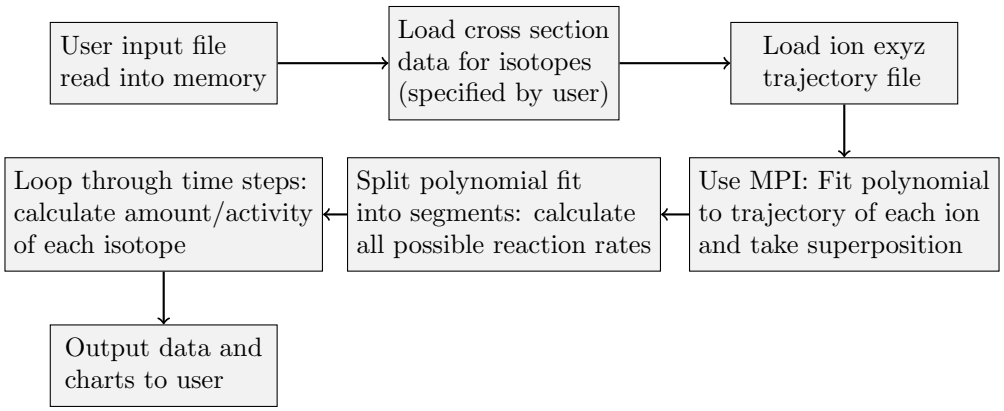


Figure 5.3: Flow chart of major processes in the Activity code

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

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

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

Chapter 6

Methodology: Interatomic Potential Fitting

Chapter Summary

Introduction

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

Math Techniques

Lagrange Polynomial Interpolation

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

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

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

Listing 6.1: Add two numbers function

```
// lagrange polynomial interpolation
function lpinterp_y(x, data)
  // x - point value being interpolated at to calc f(x)
  // data - four x,y data points (n by 2 array)
  n = len(data, 1)
  y = 0.0
  for i = 1,n
    l = 1.0
    for j = 1,n
      if (i != j) then
        l = l * (x - data[j][1]) / (data[i][1] - data[j][1])
      end if
    next j
    y = y + l * data[i][2]
  next i
end function lpinterp_y
```

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

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

$$\begin{aligned} q_n(x) &= \sum_{i=0}^{i=n} y_i g(i,x) \\ \text{where } g(i,x) &= \left(\prod_{j=0, j\neq i}^{j=n} \frac{1}{x_i - x_j} \right) \times \left(\sum_{k=0, k\neq i}^{k=n} \prod_{j=0, j\neq k, j\neq i}^{j=n} (x - x_j) \right) \end{aligned} \tag{6.3}$$

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

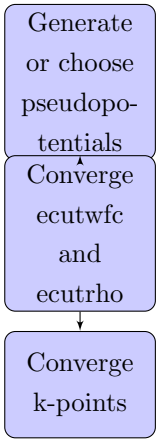
```
    fx = 1.0
    gx = 1.0
    for j = 1, n
      if (i != j) then
        fx = fx / (data[i][1] - data[j][1])
        psum = 1.0
        for k = 1, n
          if (i != k and j != k) then
            psum = psum * (x - data[k][1])
          end if
        next k
        gx = gx + psum
      end if
    next j
    dydx = dydx + fx * gx * data[i][2]
  next i
end function lpinterp_dydx
```

Splines

Simulated Annealing

Levenberg-Marquardt Optimisation

Interatomic Potential Fitting



Density Functional Theory

Quantum Espresso

DFT Calculations: Iron and Palladium

High Performance Computing

Processing and Memory Requirements

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

HECToR and Archer

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

BlueBEAR

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

Reference Database

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

32 atom randomised

108 atom defects

DFT Calibration

Choice of Pseudopotentials

Energy and Charge Density Cutoffs

K-Points

nbnd

Atomic Configurations for DFT Calculations

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

Potential Analysis and Fitting Code

Tabulated Potentials and Interpolation

Lagrange Interpolation

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

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

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

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

Simulated Annealing

Levenberg-Marquardt Optimisation

Calculating Crystal Properties

Introduction

Bulk Modulus

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

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

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

DL_POLY Contribution

Introduction

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

- Finnis Sinclair
- EAM
- EEAM

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

Modifying DL_POLY: 2BEAM

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

Mailshot Extract

Listing 6.2: Mailshot Extract

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

Continuous Optimization

Implementation of Optimization Algorithms

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

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

LMA

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

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

Chapter 7

Activity Code Development & Publication

Chapter Summary

Activation by Ion Irradiation

Computer Package Development

¡Intro¿

Chapter 8

Ab Initio Reference Database

Chapter Summary

Activation by Ion Irradiation

Computer Package Development

¡Intro¿

Chapter 9

Interatomic Potential Fitting

Chapter Summary

Activation by Ion Irradiation

Computer Package Development

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

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

Cython

F2PY

OpenMP

Development

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

Chapter 10

Molecular Dynamics

Chapter Summary

Activation by Ion Irradiation

Computer Package Development

¡Intro¿

Chapter 11

Conclusions

Chapter Summary

¡Intro¿

Restating Objectives

Contributions of this Thesis While Answering the Original Question

Chapter 12

Future Work

Chapter Summary

¡Intro¿

Appendices

Appendix A

DFT Calibration

Table A.1: Nbnd settings

	Al	Cr	Fe	Ru	Pd
Valence Electrons	3	14	16	16	18
Nbnd 1 atom	3	10	12	12	13
Nbnd 2 atoms	5	20	23	23	26
Nbnd 4 atoms	9	40	45	45	51
Nbnd 16 atoms	34	157	180	180	202
Nbnd 31 atoms	66	304	348	348	391
Nbnd 32 atoms	68	314	359	359	404
Nbnd 33 atoms	70	324	370	370	416
Nbnd 107 atoms	225	1049	1199	1199	1349
Nbnd 108 atoms	227	1059	1210	1210	1361
Nbnd 109 atoms	229	1069	1221	1221	1374
Nbnd 255 atoms	536	2499	2856	2856	3213
Nbnd 256 atoms	538	2509	2868	2868	3226
Nbnd 257 atoms	540	2519	2879	2879	3239

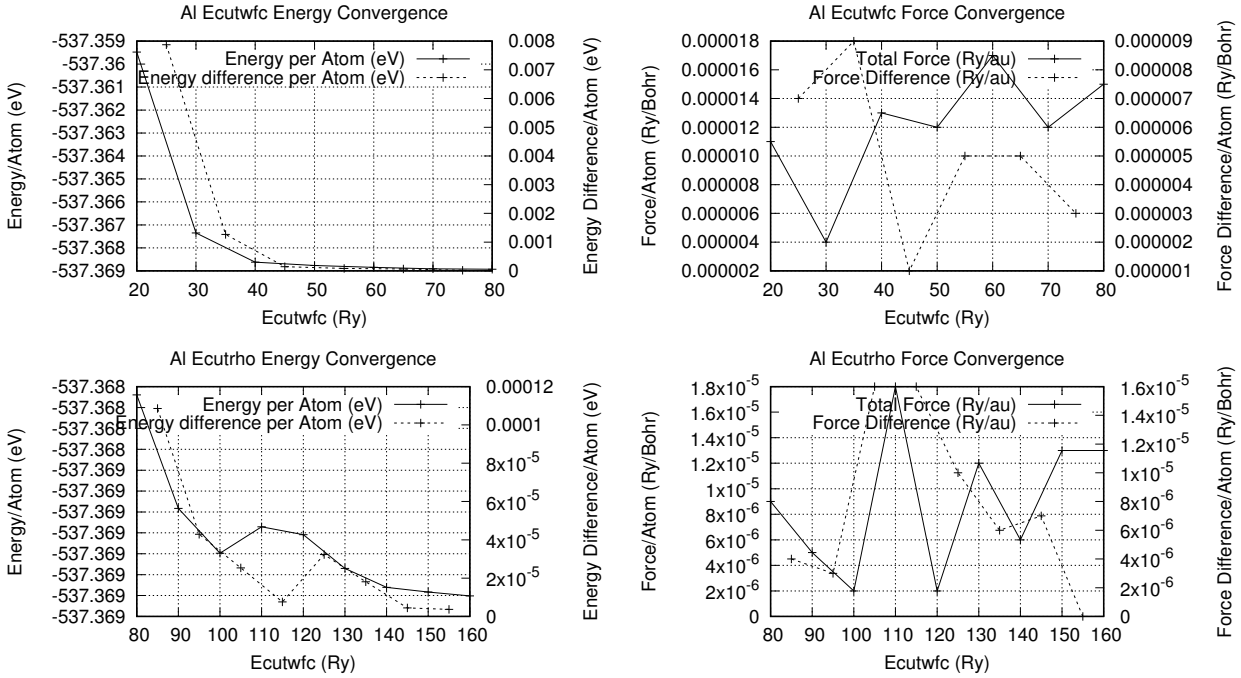


Figure A.1: Graph caption

Ecutwfc and Ecutrho Convergence

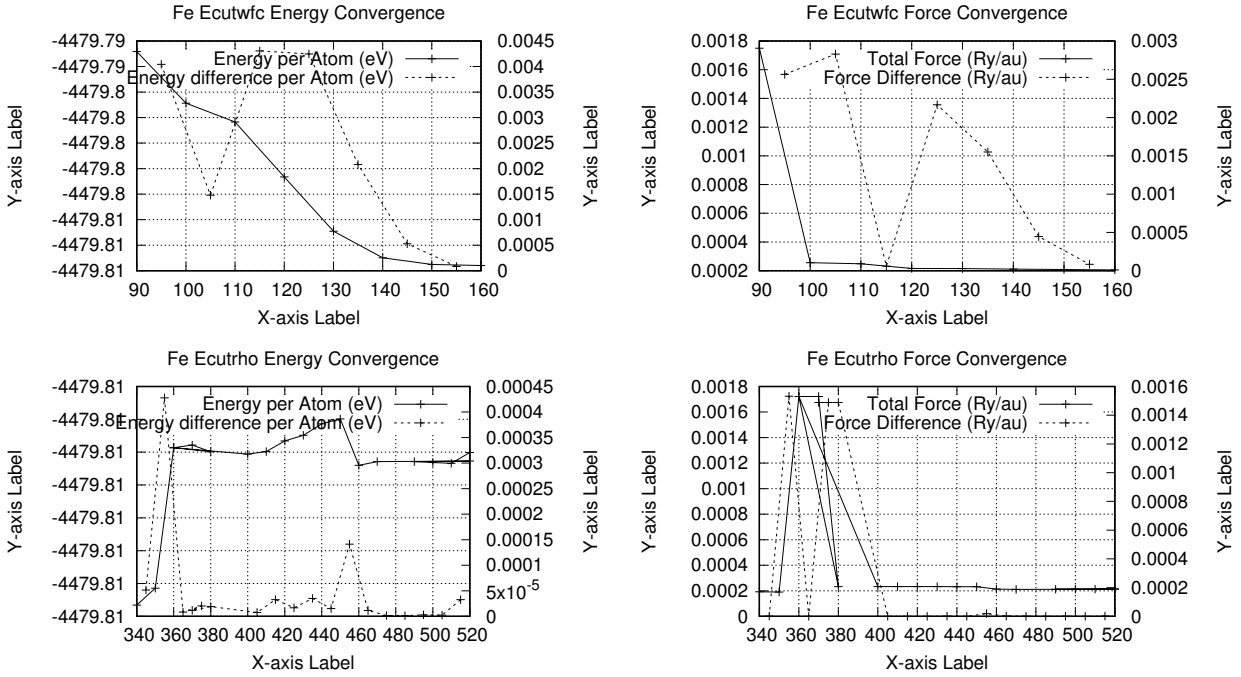


Figure A.2: Graph caption

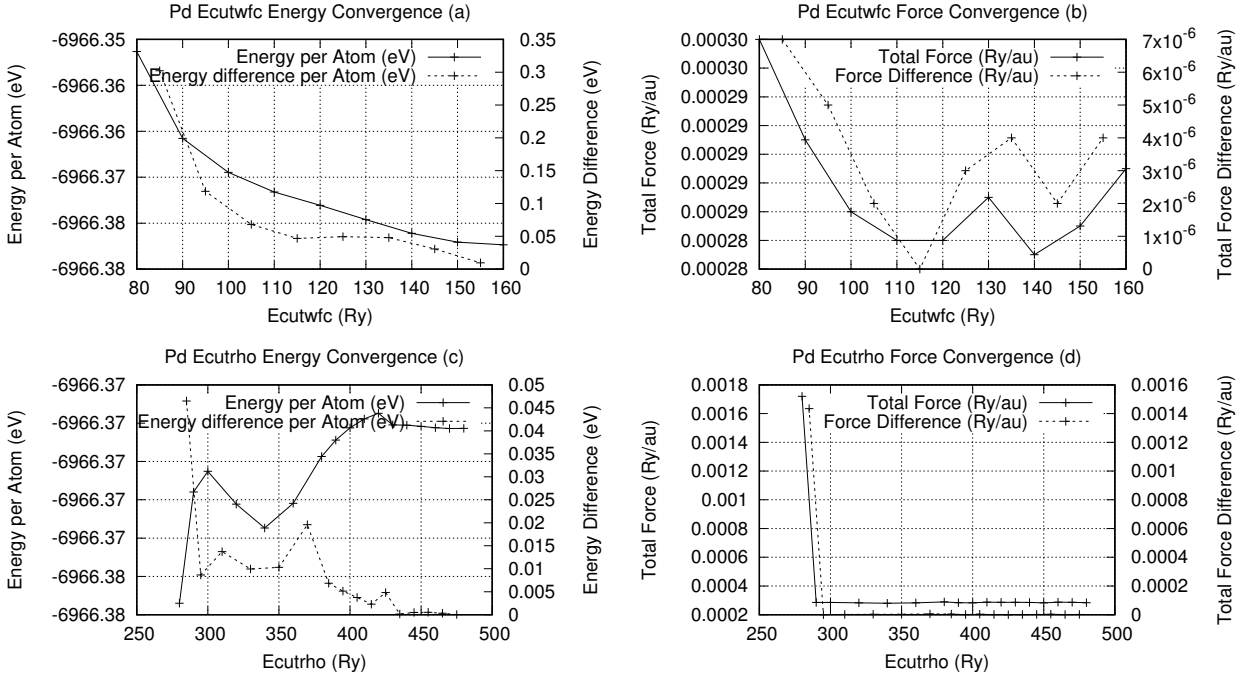


Figure A.3: Graph caption

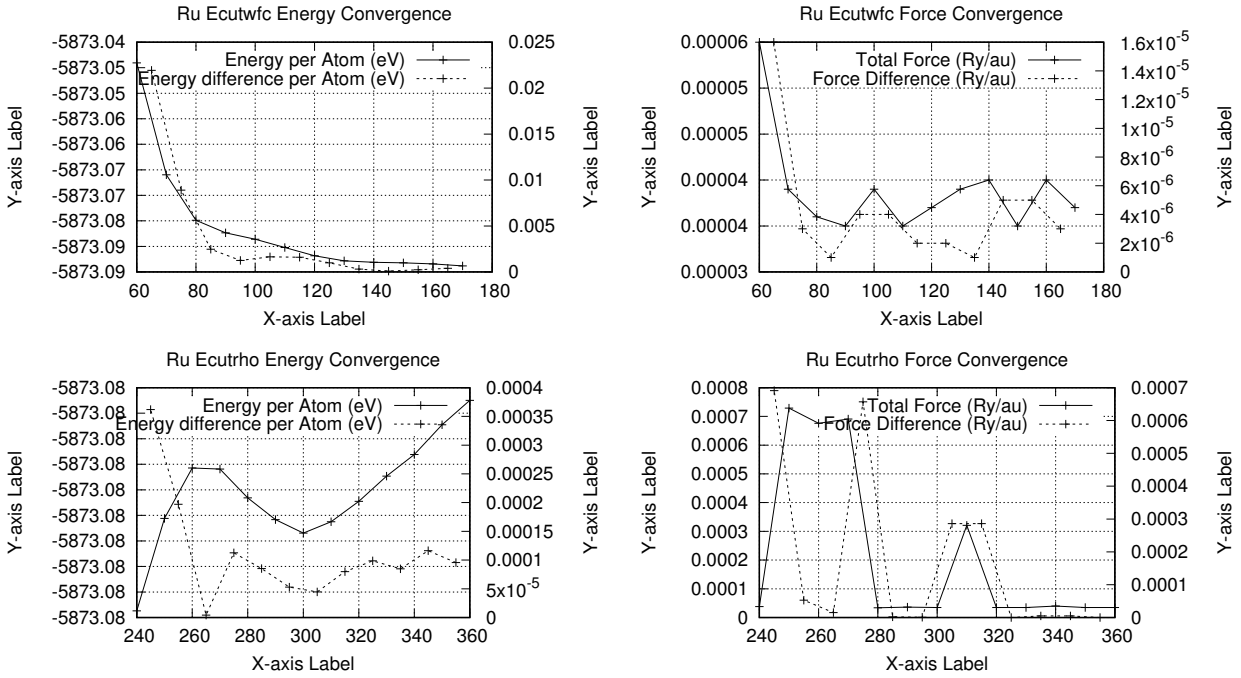


Figure A.4: Graph caption

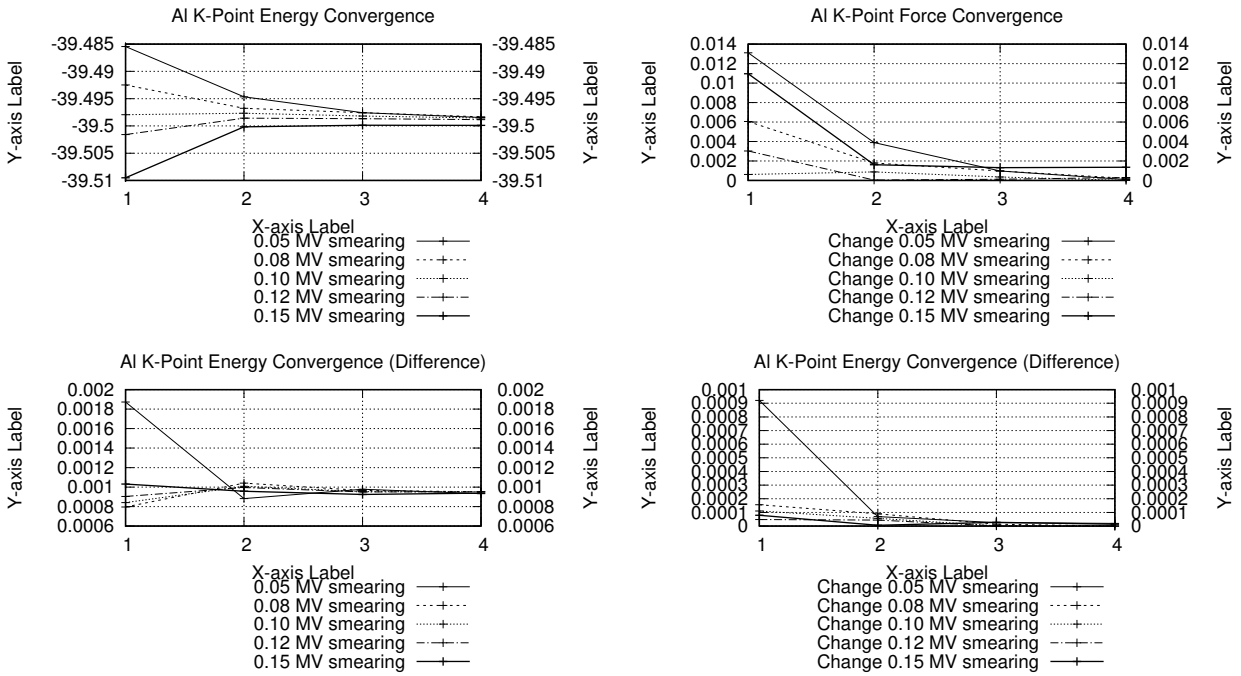


Figure A.5: Graph caption

K-point Convergence

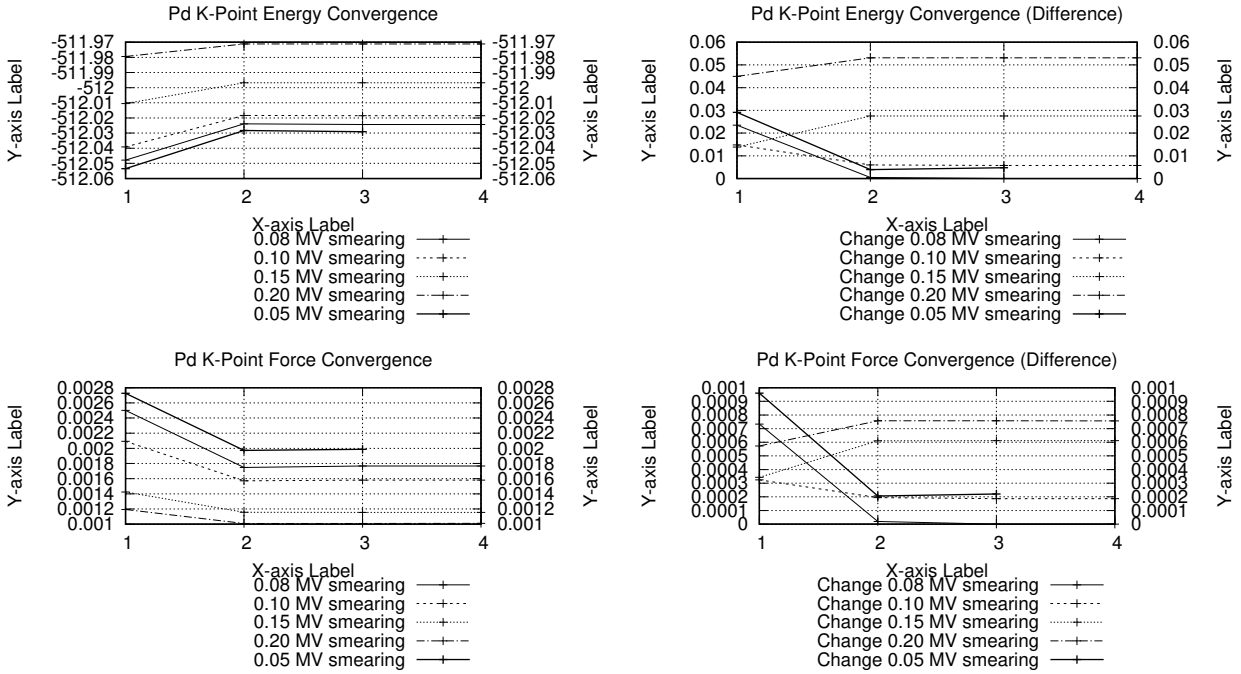


Figure A.6: Graph caption

Appendix B

Important Codes and Scripts

The contents...

Appendix C

Activity Paper published in Computer Physics Communications

Computer Physics Communications 216 (2017) 138–144

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Activity computer program for calculating ion irradiation activation☆

Ben Palmer *, Brian Connolly, Mark Read

University of Birmingham, United Kingdom

ARTICLE INFO

Article history:
Received 1 June 2015
Received in revised form 15 February 2017
Accepted 19 February 2017
Available online 29 March 2017

ABSTRACT

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

Program summary
Program title: Activity
Catalogue identifier: AFBS_v1_0
Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AFBS_v1_0.html
Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland
Licensing provisions: GNU GPL v3
No. of lines in distributed program, including test data, etc.: 688828
No. of bytes in distributed program, including test data, etc.: 71056048
Distribution format: tar.gz
Programming language: Fortran.
Computer: PCs or HPCs.
Operating system: Linux (tested on Debian).
Has the code been vectorized or parallelized?: OpenMPI
RAM: 250MB per process + 200MB overhead
Classification: 2.2, 17.8.

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

Solution method: The program loads cross section data from the TENDL database and trajectory data from a SRIM [1] simulation 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.

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

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

http://dx.doi.org/10.1016/j.cpc.2017.02.025
0010-4655/

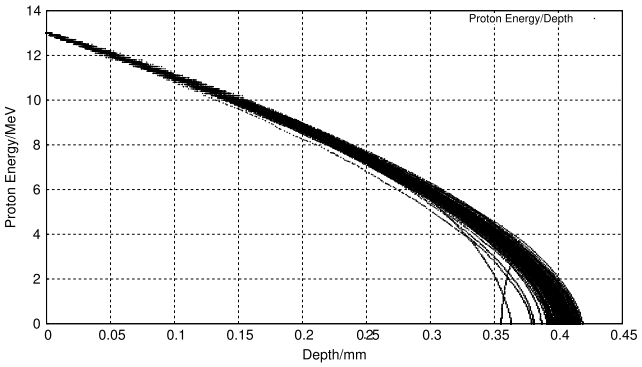


Fig. 1. One hundred simulated 13 MeV proton energy loss curves in Fe simulated with SRIM [4].

1. Background

1.1. Ion irradiation at the University of Birmingham

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

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

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

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

1.2. Simulating ion irradiation with SRIM

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

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

1.3. Transmutation of nuclei by ion irradiation

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

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

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

1.4. Radioactive decay

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

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

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:

$$N(t) = N(0) \exp(-t\lambda). \tag{3}$$

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:

$$A(t) = -N'(t) = \lambda N(t) \tag{4}$$

$$A(t) = \lambda N(0) \exp(-t\lambda). \tag{5}$$

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.

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

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].

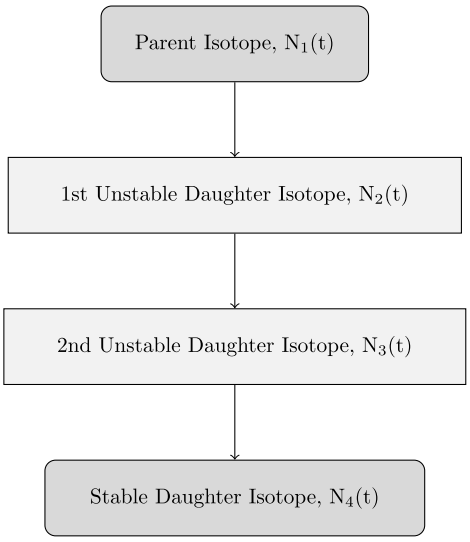


Fig. 2. An example decay chain from an unstable parent isotope, through unstable daughter isotopes ending with a stable daughter isotope.

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.

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

1.7. Constructing the differential equations

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

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

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

$$N_1'(t) = \omega_1 - \lambda_1 N_1(t) \tag{8}$$

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

$$N_z'(t) = \omega_z + b_{z-1}\lambda_{z-1}N_{z-1}(t). \tag{10}$$

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

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

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

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

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).

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

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

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

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):

$$N_4(s) = \frac{1}{(s+\lambda_1)(s+\lambda_2)(s+\lambda_3)(s+\lambda_4)}b_2b_3b_4\lambda_1\lambda_2\lambda_3N_1(0)$$

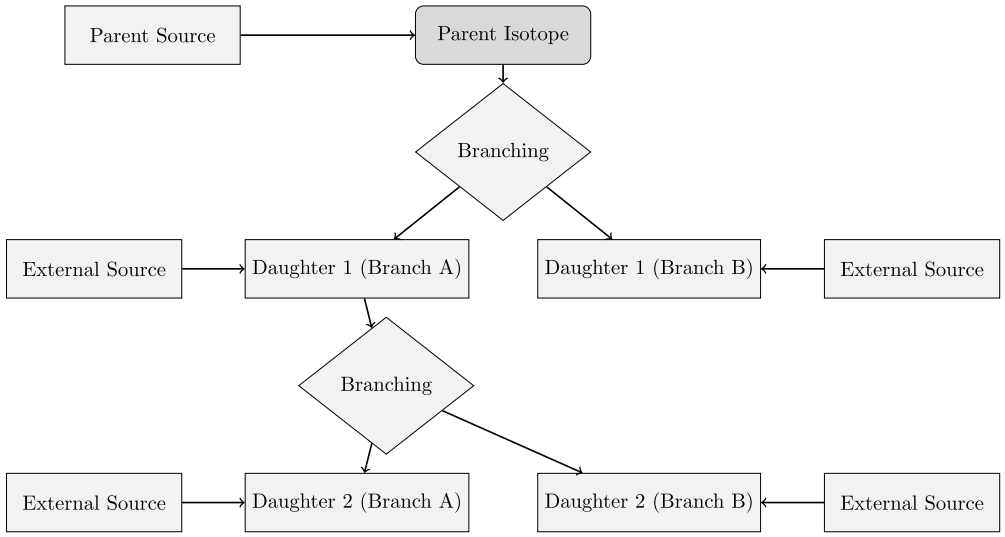


Fig. 3. An example of several decay chains including branching factors and possible external source terms for each isotope on each chain.

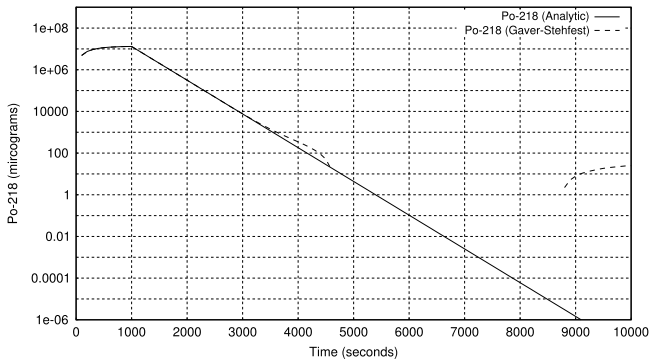


Fig. 4. Decay of Po-218: Analytic and Gaver– Stehfest Calculation [5].

$$
\begin{aligned}
&+\frac{1}{(s+\lambda_2)(s+\lambda_3)(s+\lambda_4)}b_3b_4\lambda_2\lambda_3N_2(0)\\
&+\frac{1}{(s+\lambda_3)(s+\lambda_4)}b_4\lambda_3N_3(0)+\frac{1}{(s+\lambda_4)}N_4(0)\\
&+\frac{1}{s(s+\lambda_1)(s+\lambda_2)(s+\lambda_3)(s+\lambda_4)}b_2b_3b_4\lambda_1\lambda_2\lambda_3\omega_1\\
&+\frac{1}{s(s+\lambda_2)(s+\lambda_3)(s+\lambda_4)}b_3b_4\lambda_2\lambda_3\omega_2\\
&+\frac{1}{s(s+\lambda_3)(s+\lambda_4)}b_4\lambda_3\omega_3+\frac{1}{s(s+\lambda_4)}\omega_4. \qquad (17)
\end{aligned}
$$

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

$$
\begin{aligned}
N_4(s) = &\frac{1}{s(s+\lambda_1)(s+\lambda_2)(s+\lambda_3)}b_2b_3b_4\lambda_1\lambda_2\lambda_3N_1(0)\\
&+\frac{1}{s(s+\lambda_2)(s+\lambda_3)}b_3b_4\lambda_2\lambda_3N_2(0)\\
&+\frac{1}{s(s+\lambda_3)}b_4\lambda_3N_3(0)+N_4(0)\\
&+\frac{1}{s^2(s+\lambda_1)(s+\lambda_2)(s+\lambda_3)}b_2b_3b_4\lambda_1\lambda_2\lambda_3\omega_1\\
&+\frac{1}{s^2(s+\lambda_2)(s+\lambda_3)}b_3b_4\lambda_2\lambda_3\omega_2\\
&+\frac{1}{s^2(s+\lambda_3)}b_4\lambda_3\omega_3+\frac{1}{s^2}\omega_4. \qquad (18)
\end{aligned}
$$

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.

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

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

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

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

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.

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

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

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

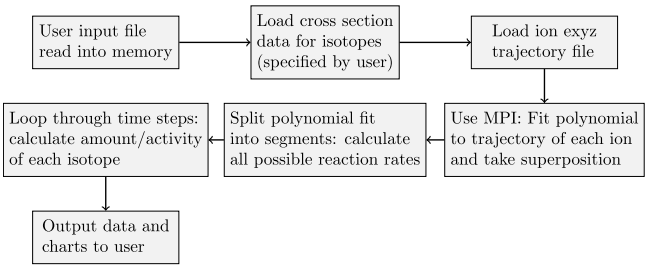


Fig. 5. Flow chart of major processes in the Activity code.

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

1.10. Preference: analytic over numeric

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

2. Computational methods

The Activity program has been developed in Fortran and takes advantage of MPI (Message 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 μA 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

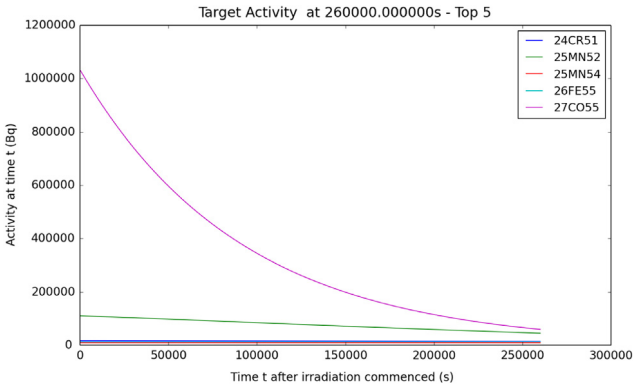


Fig. 6. Sample Activity code output chart for the top five most active isotopes for Iron irradiated by 36 MeV protons.

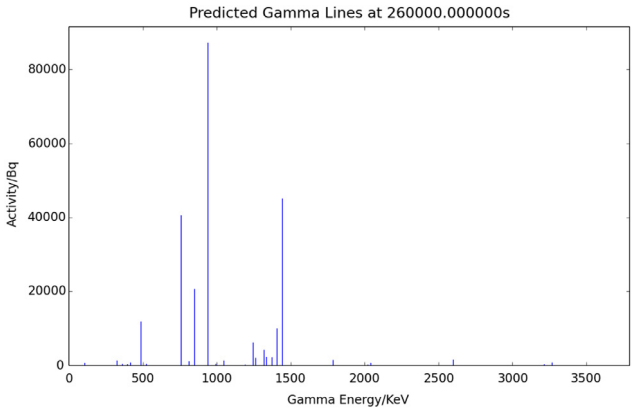


Fig. 7. Sample Activity code output chart for the expected gamma lines to be measured for Iron irradiated by 36 MeV protons.

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

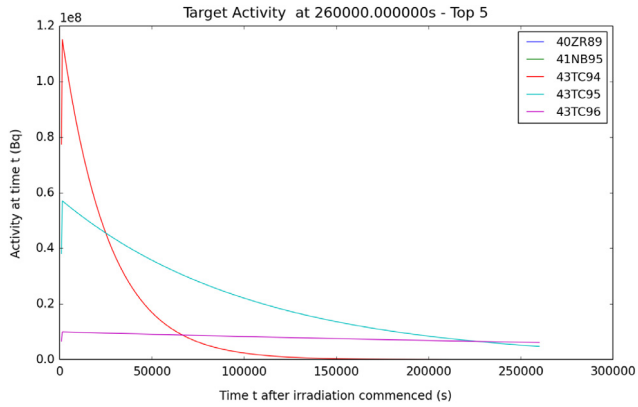


Fig. 8. Sample Activity code output chart for the top five most active isotopes for Molybdenum irradiated by 13 MeV protons.

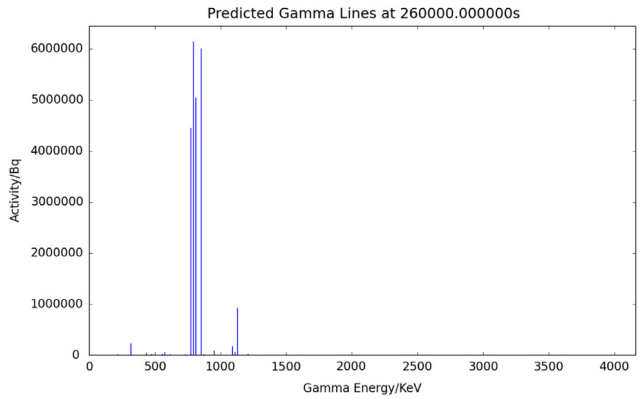


Fig. 9. Sample Activity code output chart for the expected gamma lines to be measured for Molybdenum irradiated by 13 MeV protons.

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

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

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

Acknowledgments

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

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

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

References

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

Appendix D

Activity Source Code

https://github.com/BenPalmer1983/activity

Listing D.1: Add two numbers function

```
!=============================================================================================
!
! LMA algorithm
! Levenberg, K.: 1944, Quarterly of Applied Mathematics 2, 164:
!   Dampened Newton Gauss Algorithm
! Marquardt, D.: 1963, Journal of the Society for Industrial and Applied Mathematics 11(2), 431:
!   Introduction of Lambda (Marquardt Parameter)
! R. Fletcher 1971, A modified Marquardt Subroutine for Non-linear Least Squares
!   Starting Lambda and lower lambda cutoff (where it is replaced by 0)
! M. Transtrum, J. Sethna 2012, Improvements to the Levenberg-Marquardt algorithm for nonlinear
! least-squares minimization:
!   Delayed gratification scheme for varying lambda
! Jens Jessen-Hansen 2011, Levenberg-Marquardts algorithm Project in Numerical Methods:
!   Independent/Diagonal covariance matrix for weighting
! Parameter limits implemented
! Finite difference method used to calculate J
!
!=============================================================================================

! Note:
! f2py doesn't work well with external or interface defined functions
! therefore the function is picked using an integer and all functions
! available exist in another file
!

FUNCTION LMA&
(F, parametersIn, points) &
RESULT (parametersOut)
! LMA Function
IMPLICIT NONE
! Vars: In
INTEGER :: F
REAL(8), Dimension(:) :: parametersIn
REAL(8), Dimension(:,:) :: points
! Vars: Out
REAL(8), Dimension(1:size(parametersIn,1)) :: parametersOut
! Parameters
REAL(8), Dimension(1:size(parametersIn,1), 1:2) :: limits
LOGICAL :: weighting
REAL(8) :: p_conv_threshold
REAL(8) :: rss_threshold
```

```
INTEGER :: lma_loops
INTEGER :: lambda_loops
REAL(8) :: difference_factor

weighting = .True.
p_conv_threshold = 1.0D-20
rss_threshold = 1.0D-20
lma_loops = 20
lambda_loops = 100
difference_factor = 1.0D-9
limits = 0.0D0

! Call subroutine
CALL LMA_Sub (&
F, parametersIn, points, &
limits, &
weighting, p_conv_threshold, rss_threshold, &
lma_loops, lambda_loops, difference_factor, &
parametersOut)
END FUNCTION LMA


FUNCTION LMA_L&
(F, parametersIn, points, limits) &
RESULT (parametersOut)
! LMA Function
IMPLICIT NONE
! Vars: In
INTEGER :: F
REAL(8), Dimension(:) :: parametersIn
REAL(8), Dimension(:,:) :: points
REAL(8), Dimension(:,:) :: limits
! Vars: Out
REAL(8), Dimension(1:size(parametersIn,1)) :: parametersOut
! Parameters
LOGICAL :: weighting
REAL(8) :: p_conv_threshold
REAL(8) :: rss_threshold
INTEGER :: lma_loops
INTEGER :: lambda_loops
REAL(8) :: difference_factor

weighting = .True.
p_conv_threshold = 1.0D-20
rss_threshold = 1.0D-20
lma_loops = 100
lambda_loops = 100
difference_factor = 1.0D-9

! Call subroutine
CALL LMA_Sub (&
F, parametersIn, points, &
limits, &
weighting, p_conv_threshold, rss_threshold, &
lma_loops, lambda_loops, difference_factor, &
parametersOut)
END FUNCTION LMA_L


FUNCTION LMA_Full&
(F, parametersIn, points, &
```

```
limits, &
weighting, p_conv_threshold, rss_threshold, &
lma_loops, lambda_loops, difference_factor) &
RESULT (parametersOut)
! LMA Function
IMPLICIT NONE
! Vars: In
INTEGER :: F
REAL(8), Dimension(:) :: parametersIn
REAL(8), Dimension(:,:) :: points
REAL(8), Dimension(:,:) :: limits
LOGICAL :: weighting
REAL(8) :: p_conv_threshold
REAL(8) :: rss_threshold
INTEGER :: lma_loops
INTEGER :: lambda_loops
REAL(8) :: difference_factor
! Vars: Out
REAL(8), Dimension(1:size(parametersIn,1)) :: parametersOut
! Call subroutine
CALL LMA_Sub (&
F, parametersIn, points, &
limits, &
weighting, p_conv_threshold, rss_threshold, &
lma_loops, lambda_loops, difference_factor, &
parametersOut)
END FUNCTION LMA_Full


SUBROUTINE LMA_Sub (&
F, parametersIn, points, &
limits, &
weighting, p_conv_threshold, rss_threshold, &
lma_loops, lambda_loops, difference_factor, &
parametersOut)

IMPLICIT NONE !Force declaration of all variables
! Vars: In/Out
INTEGER, INTENT(IN) :: F
REAL(8), INTENT(IN), Dimension(:) :: parametersIn
REAL(8), INTENT(IN), Dimension(:,:) :: points
REAL(8), Dimension(:,:) :: limits
LOGICAL, INTENT(IN) :: weighting
REAL(8) :: p_conv_threshold
REAL(8) :: rss_threshold
INTEGER :: lma_loops
INTEGER :: lambda_loops
REAL(8) :: difference_factor
REAL(8), Dimension(:) :: parametersOut
! Vars: Private variables
INTEGER :: i, n, k
REAL(8), Dimension(1:size(parametersIn,1)) :: parameters, parametersT, parametersL
REAL(8), Dimension(1:size(points,1),1:size(parametersIn,1)) :: J
REAL(8), Dimension(1:size(points,1)) :: R
REAL(8) :: x, y, fx, fxP, dVal, dValFactor
REAL(8) :: lambda, lambdaCutoff
REAL(8) :: rss, p_conv_value
LOGICAL :: converged
LOGICAL :: run_lma_loop, run_lambda_loop

! Init vars
```

```
parameters = parametersIn
converged = .false.

n = 1
run_lma_loop = .True.
Do While((n .LE. 50) .AND. (run_lma_loop .EQV. .True.)) ! maximum 50 loops
  Call LMA_Make_J(F, parameters, points, difference_factor, J, R, rss)
! Starting lambda
  If(n.eq.1)Then
    lambdaCutoff = LMA_Lambda(J)
    lambda = lambdaCutoff
  End If

! Start loop - keep increasing lambda if not an improvement
  k = 1
  run_lambda_loop = .True.
  Do While((k .LE. 100) .AND. (run_lambda_loop .EQV. .True.)) ! max 100
    Call LMA_Loop(F, parameters, points, limits, weighting, J, R, rss, lambda, lambdaCutoff, &
    p_conv_threshold, p_conv_value, run_lambda_loop)
    k = k + 1
  End Do
  If((rss .LE. rss_threshold) .OR. (p_conv_value .LE. p_conv_threshold))Then
    run_lma_loop = .False.
  End If
  n = n + 1
End Do
parametersOut = parameters
END SUBROUTINE LMA_Sub


SUBROUTINE LMA_Make_J(F, parameters, points, difference_factor, J, R, rss)
! Makes J matrix finite difference
! R matrix
INTEGER :: F
REAL(8), Dimension(:) :: parameters
REAL(8), Dimension(:,:) :: points
REAL(8) :: difference_factor
REAL(8), INTENT(OUT), Dimension(1:size(points,1),1:size(parameters,1)) :: J
REAL(8), INTENT(OUT), Dimension(1:size(points,1)) :: R
REAL(8), INTENT(OUT) :: rss

INTEGER :: i, k
REAL(8) :: fx, fxP, finite_difference
REAL(8), Dimension(1:size(parameters,1)) :: parameters_perturbed

rss = 0.0D0
Do i=1,size(points,1)
  ! Calculated fx
  fx = LMA_F(F, parameters, points(i,1))
  R(i) = fx - points(i,2)
  rss = rss + R(i)**2
  ! Add difference to each parameter to find deriv wrt each parameter
  Do k=1,size(parameters,1)
    finite_difference = abs(parameters_perturbed(k) * difference_factor)
    If (finite_difference .EQ. 0.0D0) Then
      finite_difference = 1.0D-9
    End If
    parameters_perturbed = parameters
    parameters_perturbed(k) = parameters(k) + finite_difference
    fxP = LMA_F(F, parameters_perturbed, points(i,1))
    J(i,k) = (fxP-fx)/finite_difference
```

```
  End Do
End Do
END SUBROUTINE LMA_Make_J


SUBROUTINE LMA_Loop(F, parameters, points, limits, weighting, J, R, rss, lambda, &
lambdaCutoff, p_conv_threshold, p_conv_value, runloop)
IMPLICIT NONE
INTEGER :: F
REAL(8), INTENT(INOUT), Dimension(:) :: parameters
REAL(8), Dimension(:,:) :: points
REAL(8), Dimension(:,:) :: limits
LOGICAL :: weighting
REAL(8), Dimension(:,:) :: J
REAL(8), Dimension(:) :: R
REAL(8) :: rss
REAL(8) :: lambda, lambdaCutoff
REAL(8) :: p_conv_threshold
REAL(8) :: p_conv_value
LOGICAL, INTENT(OUT) :: runloop

! Vars: Private variables
REAL(8), Dimension(1:size(parameters,1)) :: parameters_last
REAL(8) :: rssTrial
LOGICAL :: converged
! Delayed gratification scheme
runloop = .True.
! Store last set of parameters
parameters_last = parameters
! calculate new parameters
parameters = LMA_Calc(J,R,lambda,lambdaCutoff,parameters,limits, weighting)

! Check parameter convergence
p_conv_value = LMA_ParameterConvergence(parameters, parameters_last)

! Calc RSS
rssTrial = LMA_FUNCTION_RSS(F, points, parameters)
! Breakout if NaN
If (Isnan(rssTrial)) Then
  runloop = .False.
  parameters = parameters_last
Else
  If(p_conv_value .LT. p_conv_threshold)Then
    runloop = .False.
  ! Delayed gratification scheme - 1.5*lambda or 0.2*lambda
  Else
    If(rssTrial.gt.rss)Then ! If worse
      If(lambda.lt.lambdaCutoff)Then
        lambda = 2.0D0*lambdaCutoff
      End If
      lambda = lambda * 1.5D0
      parameters = parameters_last
    Else ! If better
      lambda = lambda * 0.2D0
      runloop = .False.
    End If
  End If
End If
END SUBROUTINE LMA_Loop


```

```
FUNCTION LMA_ParameterConvergence(p_old, p_new) RESULT (convergence)
REAL(8), Dimension(:) :: p_old
REAL(8), Dimension(:) :: p_new
REAL(8) :: convergence
INTEGER :: n
convergence = 0.0D0
Do n = 1, size(p_old, 1)
  convergence = convergence + abs(p_old(n) - p_new(n))
End Do
convergence = convergence / n
END FUNCTION LMA_ParameterConvergence

FUNCTION LMA_Calc(J,R,lambda,lambdaCutoff, parametersIn,limits,weighting) RESULT (parametersOut)
IMPLICIT NONE !Force declaration of all variables
! In
REAL(8), Dimension(:,:) :: J                              ! Jacobian
REAL(8), Dimension(:) :: R                                ! Residuals
REAL(8), Dimension(:) :: parametersIn
REAL(8), Dimension(:,:) :: limits
REAL(8) :: lambda
REAL(8) :: lambdaCutoff
LOGICAL :: weighting
! Out
REAL(8), Dimension(1:size(parametersIn,1)) :: parametersOut
! Private variables
INTEGER :: i
REAL(8), Dimension(1:size(R),1:size(R)) :: W          !
REAL(8), Dimension(1:size(J,2),1:size(J,1)) :: JT         ! Transpose Jacobian
REAL(8), Dimension(1:size(J,2),1:size(J,1)) :: JTW        !
REAL(8), Dimension(1:size(J,2),1:size(J,2)) :: JTWJ
REAL(8), Dimension(1:size(J,2),1:size(J,2)) :: JTWJ_Diag
REAL(8), Dimension(1:size(J,2)) :: JTWR
REAL(8), Dimension(1:size(J,2),1:size(J,2)) :: A ! Left side
REAL(8), Dimension(1:size(J,2)) :: B         ! Right side
REAL(8), Dimension(1:size(J,2)) :: P                      ! Change
LOGICAL :: weighting_default
! ***********
! P = (JTJ+Lambda*diag(JTJ))^(-1)(-1*JTR)
! (JTJ+Lambda*diag(JTJ)) P = (-1*JTR)
! ***********
! Check Marquardt parameter cutoff
! Check Marquardt parameter cutoff
If(lambda.lt.lambdaCutoff)Then
  lambda = 0.0D0
End If
! Prep covariance matrix
W = 0.0D0
If (weighting)Then
  weighting_default = .False.
  Do i = 1, size(R)
    If (R(i) .LT. 1.0D-50) Then
      weighting_default = .True.
      Exit
    End If
  End Do
  ! Use default weighting matrix as R it becoming too small
  If (weighting_default) Then
    Do i=1,size(R)
      W(i,i) = 1.0D99
    End Do
  ! Use calculated weighting
```

```
  Else
    Do i=1,size(R)
      W(i,i) = 1.0D0/R(i)**2
    End Do
  End If
Else
  ! No weighting
  Do i=1,size(R)
    W(i,i) = 1.0D0
  End Do
End If

! Matrix calculations
JT = Transpose(J)   ! matrix.f90
JTW = matmul(JT,W)
JTWJ = matmul(JTW,J)
JTWR = matmul(JTW,R)
JTWJ_Diag = lambda*DiagMatrix(JTWJ) ! Dampening Matrix
A = MatAdd(JTWJ,JTWJ_Diag)
B = -1.0D0*JTWR
P = lss(A, B)

! Update parameters
Do i=1,size(P)
  If (limits(i,1) .LE. limits(i,2)) Then
    parametersOut(i) = parametersIn(i) + P(i)
    If (limits(i,1) .LT. limits(i,2)) Then
      If(parametersOut(i) .LT. limits(i,1))Then
        parametersOut(i) = limits(i,1)
      Else If(parametersOut(i) .GT. limits(i,2))Then
        parametersOut(i) = limits(i,2)
      End If
    End If
  End If
End Do
END FUNCTION LMA_Calc


FUNCTION LMA_Lambda(J) RESULT (lambda)
IMPLICIT NONE !Force declaration of all variables
! In
REAL(8), Dimension(:,:) :: J                              ! Jacobian
! Out
REAL(8) :: lambda
! Private variables
REAL(8), Dimension(1:size(J,2),1:size(J,1)) :: JT         ! Transpose Jacobian
REAL(8), Dimension(1:size(J,2),1:size(J,2)) :: JTJ, JTJ_Inv
REAL(8) :: traceJTJ
! Calculate starting lambda/lambda cutoff
JT = Transpose(J)
JTJ = matmul(JT,J)
JTJ_Inv = InvertMatrix(JTJ)
! Check Marquardt parameter cutoff
traceJTJ = Trace(JTJ_Inv)
lambda = traceJTJ**(-1)
END FUNCTION LMA_Lambda


FUNCTION LMA_FUNCTION_RSS(F, points, parameters) RESULT (rss)
! RSS between function and calculated points
IMPLICIT NONE
```

```
408 ! Vars: In
409 INTEGER :: F
410 REAL(8), Dimension(:,:) :: points
411 REAL(8), Dimension(:) :: parameters
412 ! Out
413 REAL(8) :: rss
414 ! Private variables
415 REAL(8) :: x, y, fx
416 INTEGER :: i
417 ! RSS
418 rss = 0.0D0
419 Do i=1,size(points,1)
420   x = points(i,1)
421   y = points(i,2)
422   fx = LMA_F(F, parameters, x)
423   rss = rss + (y - fx)**2
424 End Do
425 END FUNCTION LMA_FUNCTION_RSS
426
427
428 FUNCTION LMA_F(F, parameters, x) RESULT (y)
429 IMPLICIT NONE
430 INTEGER :: F
431 REAL(8), DIMENSION(:) :: parameters
432 REAL(8) :: x
433 REAL(8) :: y
434 y = F_Function(F, parameters, x)
435 END FUNCTION LMA_F
```

Appendix E

Useful Constants, Conversion Factors

Useful Fundamental Constants

Table E.1: Useful Fundamental Constants

Constant	Value
Bohr Magneton μ_B	$9.274 * 10^{-24} JT^{-1}$
Nuclear Magneton μ_n	$-9.662 * 10^{-27} JT^{-1}$

Conversion Factors

Table E.2: Useful Conversion Factors

1.0 eV	0.0735 Ry
1.0 Ry	13.606 eV
1.0 Ry/Bohr3	1.47105x104 GPa
1.0 GPa	6.798x10-5 Ry/Bohr3
1.0 bar	1.0 5 Pa
1.0 a	1.01325 5 Pa

Bibliography

[1] H. Khartabil. *Gen-4 Reactors*. 2013. URL: http://www.gen-4.org/GIF/About/documents/25-Session2-3-Khartabil.pdf (visited on 05/20/2013).

[2] MIT Open Courseware. *Engineering of Nuclear Systems - Lecture 6A*. 2013. URL: http://ocw.mit.edu/courses/nuclear-engineering/22-06-engineering-of-nuclear-systems-fall-2010/lectures-and-readings/MIT22_06F10_lec06a.pdf (visited on 05/20/2013).

[3] MIT Open Courseware. *Engineering of Nuclear Systems - Lecture 6B*. 2013. URL: http://ocw.mit.edu/courses/nuclear-engineering/22-06-engineering-of-nuclear-systems-fall-2010/lectures-and-readings/MIT22_06F10_lec06b.pdf (visited on 05/20/2013).

[4] E. Malambu A. Orden D. Struwe P. Agostini S. Monti A. Alemberti J. Carlsson. "European lead fast reactor - ELSY". In: *Nuclear Engineering and Design* 241 (2011), pp. 3470–3480.

[5] N J Carron. *An Introduction to the Passage of Energetic Particles through Matter*. 2007. ISBN: 978-0-7503-0935-6.

[6] World Nuclear. 2018. URL: http://www.world-nuclear.org/information-library/non-power-nuclear-applications/radioisotopes-research/research-reactors.aspx (visited on 11/08/2018).

[7] IAEA. 2018. URL: https://nucleus.iaea.org/RRDB/RR/ReactorSearch.aspx (visited on 11/08/2018).

[8] ORNL. 2018. URL: https://neutrons.ornl.gov/hfir/parameters (visited on 11/10/2018).

[9] 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.

[10] 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.

[11] Graeme J. Ackland. "Two-band second moment model for transition metals and alloys". In: *Condensed Materials* (2005).

[12] J. B. Adams F. Ercolessi. "Interatomic Potentials from First-Principles Calculations: the Force-Matching Method". In: *Europhys. Lett.* 26 (8 1994), pp. 583–588.

[13] A. Cadien T. Fujita H. W. Sheng M. J. Kramer and M. W. Chen. "Highly Optimized Embedded-Atom-Method Potentials for Fourteen FCC Metals". In: *Europhys. Lett.* 26 (8 1994), pp. 583–588.

[14] F. D. Murnaghan. "The Compressibility of Media Under Extreme Pressures". In: (1944).

[15] F. Birch. "Finite elastic strain of cubic crystals". In: (1947).

[16] Jiechao Cui Baoqin Fu Min Li Qing Hou. "Molecular dynamics simulations of tungsten bombardments on tungsten". In: *Nuclear Inst. and Methods in Physics Research B* (2018), In Press.

[17] H. Stehfest. "Algorithm 368: Numerical Inversion of Laplace Transform". In: *Communications of the ACM* 13 (1970), pp. 47–49.