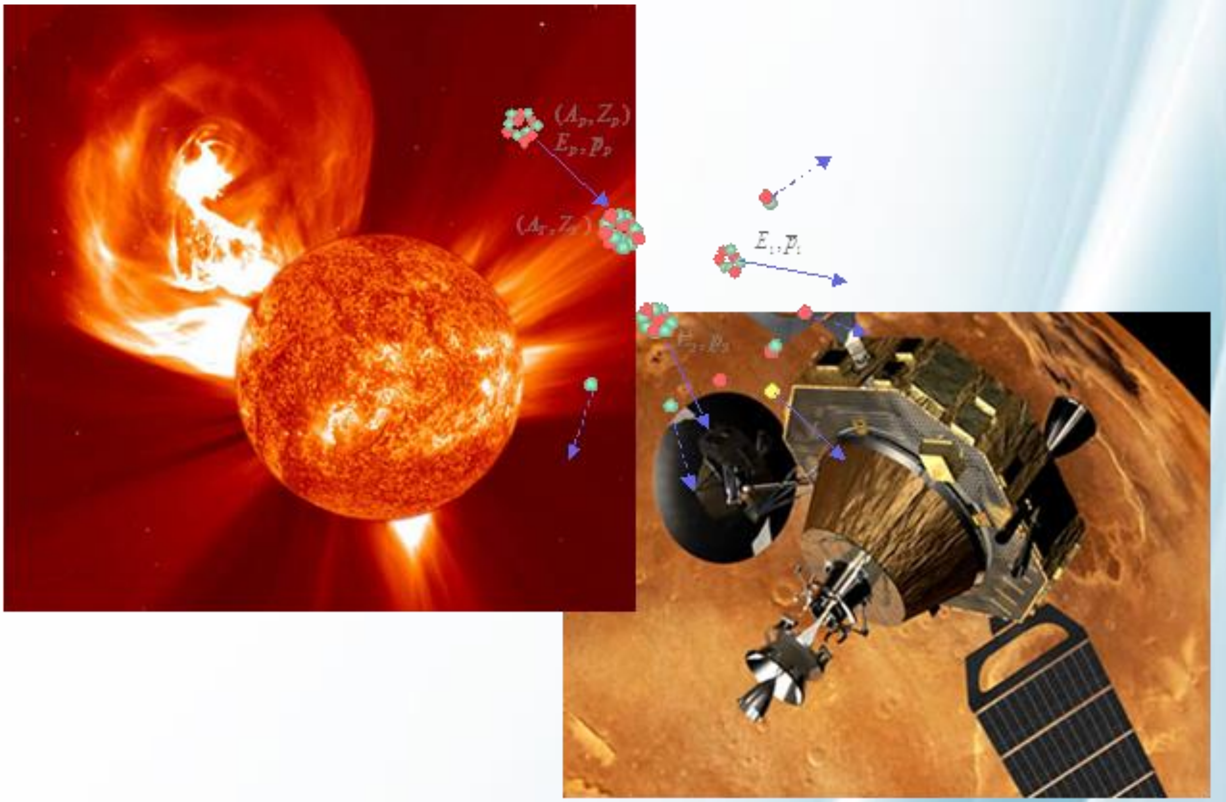
Kallisto Consultancy



ESHIEM Project Ion Rapid 1D Shielding Simulation Software (IRONSSIS)

Authors: **Pete Truscott**

Reference: **KALLISTO/TN/14028**

Issue/Revision: **1.0 / Rev 0**

Date: **February 2015**

EUROPEAN SPACE AGENCY
CONTRACT REPORT
The work described in this report was untaken within ESA contract 4000107025/12/NL/GLC.
Responsibility for the contents resides in the author or organisation that prepared it.

PROVISIONAL RELEASE CONDITIONS

This document is approved for distribution within the ESHIEM Contract Project Team and associated ESA staff only. Wider distribution is not permitted without prior authorisation by Kallisto Consultancy Ltd

European Space Agency Report Documentation Page

ESA STUDY CONTRACT REPORT

No ESA Study Contract Report will be accepted unless this sheet is inserted at the beginning of each volume of the Report.

ESA Contract No: ***4000107025/12/NL/ GLC***	SUBJECT: ***Energetic Solar Heavy Ion Environment Models (ESHIEM)***	CONTRACTOR: ***Kallisto Consultancy Ltd***
* ESA CR( )No:	No. of Volumes:***1*** This is Volume No ***1***	CONTRACTOR'S REFERENCE: ***KALLISTO/TN/14028*** ***Issue 1.0, Revision 0***

ABSTRACT:

As part of ESA Contract 4000107025/12/NL/GLC supported by the Technology Research Programme, a consortium led by Kallisto Consultancy is developing new solar heavy ion and associated shielding models which may be used by spacecraft engineers to predict the radiation threat to future spacecraft mission. The objective is to provide a statistical model covering species from α-particles to uranium ions, and which will become part of an updated Solar Energetic Particle Environment Modelling (SEPEM) System. One of new tools being introduced, IRONSSIS, allows the calculation of shielding effects against incident ions more efficiently than MULASSIS, which is currently used by SEPEM. This document describes the design and function of the new software.

The work described in this report was done under ESA Contract. Responsibility for the contents resides in the author or organisation that prepared it.

Names of authors:
Pete Truscott

** NAME OF ESA STUDY MANAGER: ***Dr Alain Hilgers*** DIV: ***EES*** DIRECTORATE: ***TEC***	** ESA BUDGET HEADING: ***Technology Research Programme***

* Sections to be completed by ESA
** Information to be provided by ESA Study Manager

Administration page

Customer Information	
Document reference	KALLISTO/DDD/14028
Project title	Energetic Solar Heavy Ion Environment Models (ESHIEM)
Customer Organisation	ESA – European Space Technology Centre
Customer contact	Dr Piers Jiggens, ESA/ESTEC TEC-EES Dr Alain Hilgers, ESA/ESTEC TEC-EES
Contract number	ESA Contract 4000107025/12/NL/GLC
Project document reference	ESHIEM TN2B (D12)
Milestone number	M2
Date due	November 2014 (DPMR Final Review)

Author list	**Telephone and E-mail**
Dr Pete Truscott	+44 (0)1276 36350
Farnborough, Hampshire, GU14 9AJ, UK	PRTruscott@kallistoconsultancy.com

Record of changes

Issue	Date	Detail of Changes
0.A	23/02/2014	Draft
0.B	14/05/2014	Updated draft
0.C	08/10/2014	Updated draft following further implementation of S/W + feedback from ESA
0.D	28/10/2014	Major updates following modification to S/W
0.E	14/10/2014	Updated draft with all latest S/W design and validation/verification information.
1.0	14/02/2015	Formal release

Cover page image courtesy of NASA and ESA Multimedia Gallery

This page is intentionally blank.

Abstract

As part of ESA Contract 4000107025/12/NL/GLC supported by the Technology Research Programme, a consortium led by Kallisto Consultancy is developing new solar heavy ion and associated shielding models which may be used by spacecraft engineers to predict the radiation threat to future spacecraft mission. The objective is to provide a statistical model covering species from α-particles to uranium ions, and which will become part of an updated Solar Energetic Particle Environment Modelling (SEPEM) System. One of new tools being introduced, IRONSSIS, allows the calculation of shielding effects against incident ions more efficiently than MULASSIS, which is currently used by SEPEM. This document describes the requirements, design and function of the new software.

List of contents

1 Introduction

1.1 Contractual

This report has been issued by Kallisto Consultancy Ltd and the ESHIEM Project Team for ESA/ESTEC under contract 4000107025/12/NL/GLC (Technology Research Programme). This document is delivered as part of Milestone 2 and Work Package WP4000 (Deliverable D8, Technical Note TN2B), and provides the description of the implementation of the ion fast shielding analysis tool.

1.2 Background

The ionising radiation environment in space is a significant limiting factor affecting the performance, reliability and ultimately the lifetime of all spacecraft and their systems. The principal sources of radiation of concern for space missions include protons and heavier ions from galactic cosmic radiation (GCR), and electrons and light ions trapped within planetary magnetic fields, such as Earth's Van Allen belts. In addition, particle emissions from the Sun also give rise to significant direct enhancements to the radiation environment. The stochastic nature of the events, in terms of time of occurrence, spectrum, intensity, duration and particle composition, makes solar particle events (SPEs) extremely challenging to model. Yet these considerations are important drivers impacting on the design and operation of both future unmanned as well as manned missions:

- Cumulative damage which can result to systems, in particular displacement damage to solar arrays, will give rise to sudden diminution in performance potentially leading to parametric or functional failure. Design margins must therefore be introduced to ensure system operation until the end-of-life of the spacecraft, even though the occurrence, energy and magnitude of SEPEs are probabilistic in nature.
- Since protons and heavier ions can give rise to single event effects (SEEs), spacecraft microelectronics and sensors must be designed to operate reliably whilst exposed to the maximum particle flux rates expected during an SPE.
- The physical design of spacecraft and habitats which may be used for future manned missions must clearly be such as to ensure crew exposure to ionising radiation remains within national or internationally-agreed limits, and indeed minimised in accordance with the ALARA principle. In addition, the threat of SPEs also applies operational constraints on future missions (*e.g.* the timing, duration of and distances at which EVAs may be performed), and the wider infrastructure – particularly hazard alert systems – that will be needed for human interplanetary and planetary missions.

Much of the emphasis of solar energetic particle model development has been on quantifying the proton environment since these particles are emitted with the greatest fluxes. However, the effects of heavy ions are still very important and inadequately addressed to date. The relative biological effectiveness of these ions means they contribute disproportionately to the equivalent dose in tissue and effective dose in humans. Most modern radiation-tolerant and radiation-hard microelectronics used in spacecraft are selected and screened so they do not suffer failure in space due to protons. As atomic number, Z, of the ion increases, the particle unrestricted stopping power scales approximately as Z^2 for a given particle energy per unit mass. The more intense ionisation tracks which result increase the likelihood of SEEs, including catastrophic SEEs.

Within the ESA Energetic Solar Heavy Ion Environment Models (ESHIEM) Project, a consortium comprising Kallisto Consultancy, DH Consultancy, RadMod Research and TRAD is developing new solar heavy ion (HI) models to allow more accurate prediction of the potential environments from solar energetic particle events and their effects on systems and crews [1][2].

1.3 Purpose

This document covers the requirements, design and function of the new software to perform rapid calculation of shielding effects against incident ions and which will be used as an alternative to MULASSIS in the SEPEM system. The design is cross-referenced to defined requirements, and information about the validation approach and results is provided.

1.4 Structure and scope

The remaining sections of this document cover the following subjects:

- Chapter 2 describes the original user requirements for the new shielding tool, based on the URD generated as part of WP1000 of the project [14].
- Chapter 3 provides the interpretation of the User Requirements into Software Requirements.
- Chapter 4 describes the overall assumptions for the calculation process and the basic sources of data and formulae.
- Chapter 5 gives the command structure for the new tool, relevant to interfacing the tool to the updated SEPEM system.
- Chapter 6 provides a description of the components of the IRONSSIS software, including new classes and modifications made to MULASSIS, as well as the directory file structure.
- Chapter 7 describes the processes to compile, build and run the software.
- Chapter 8 outlines the validation and verification approach for the new software.
- Chapter 9 gives the results of the validation of the algorithm and the new software.
- Chapter 10 concludes by providing the current software configuration control information for the new software.

2 Background and User Requirements for IRONSSIS

In order to treat the effects of physical shielding on incident particles, the current SEPEM system uses the Geant4-based MULASSIS software [12][13]. This provides a comprehensive treatment of relevant physical processes, in particular treating the ionisation by charged particles and also nuclear interactions. However, as a result of the detailed Monte Carlo solution used by MULASSIS, lengthy simulations are required to achieve the necessary statistical significance. Whilst this has been acceptable for treating just incident protons or a few ion species, the long simulation times are likely to make MULASSIS an impractical tool for dealing with many ion species, *e.g.* from helium to iron. It is necessary to consider including approximate but faster shielding analysis tools in SEPEM as an alternative to MULASSIS.

The following tables (from reference [14]) provides a reminder of the user requirements for treating shielding against ions, and particularly fast shielding analysis for incident ions (originally designated as the Fast Physical Shielding Analysis Model or FPSAM).

The requirements are identified in two priorities:

1. High priority (considered essential for SEPEM ion analysis);
2. Moderate to low priority.

Table 1: Requirements for Physical Shielding Analysis Model.

UR ID	Description of Requirement	Importance	References & Notes
UR 31	There shall be a model to allow calculation of the effects of material shielding on the incident particle flux / fluence for simple 1D shields	1	
UR 32	The default PSAM shall be MULASSIS	1	
UR 33	As an alternative to MULASSIS, there shall be a Fast Physical Shielding Analysis Model (FPSAM), which uses approximate shielding analysis techniques to estimate particle propagation in 1D shields for heavy ions.	1	
UR 34	The FPSAM shall be relevant to HI propagation in (UR 34.1) a range of standard spacecraft materials (Al, Ti, Fe, SiO_2, Si, CFRP, Kapton) (UR 34.2) any material	 1 2	
UR 35	The FPSAM shall estimate as a function of depth and particle species: (UR 35.1) HI fluxes or fluences (UR 35.2) total ionising dose (UR 35.3) total non-ionising dose in Si and GaAs (UR 35.4) linear energy transfer (LET) spectrum in Si NOTE: TNID in Si and GaAs will be possible for protons based on internal NIEL coefficients, but in order to calculate the TNID for heavier ions, the user must provide their own coefficients.	 1 1 2 1	
UR 36	The FPSAM shall provide estimates of the uncertainties in the shielding calculation process and allow error propagation.	1	Orig. SEPEM UR14 & UR22

UR ID	Description of Requirement	Importance	References & Notes
UR 37	The user shall be able to provide modified estimates of the errors in the model, which can then be propagated.	1	Orig. SEPEM UR14 & UR22

In addition to the above IRONSSIS-specific user requirements, the following are also applicable.

Table 2: Interface requirements.

UR ID	Description of Requirement	Importance	References & Notes
UR 41	The interfaces shall be consistent with those already employed for: (UR 41.1) SEPEM (UR 41.2) SPENVIS, *i.e.* output files shall conform to SPENVIS .CSV file format to enable easy interfacing with SPENVIS and data visualisation	1	Orig. SEPEM UR24
UR 42	It shall be possible to use the HI predictions for the unshielded and shielded environments with existing SEPEM tools	1	Orig. SEPEM UR15, UR16, UR17 & UR18

Table 3: Operational environment and maintenance.

UR ID	Description of Requirement	Importance	References & Notes
UR 44	New software and data will be consistent with the design approach and software practices adopted for the existing SEPEM system.	1	
UR 45	Any new software based on Geant4 shall be consistent with existing Geant4 code in terms of modularisation and coding practice	1	
UR 46	The software shall be maintained for one year following acceptance by ESA.	1	

Table 4: Verification and validation requirements.

UR ID	Description of Requirement	Importance	References & Notes
UR 47	Verification and validation shall take place in accordance with ECSS-E-40A.	1	
UR 48	Validation of environment models shall include comparisons with: (UR 48.1) results from experiment (UR 48.2) results from other SEP models (UR 48.3) results from other effects tools	1	Orig. SEPEM UR21

3 Software Requirements Analysis

3.1 Introduction

This chapter provides a summary of the Software Requirements for the Fast Physical Shielding Analysis Model (FPSAM) based on the User Requirements given in the previous section. It narrows down some of the design options which were not fully addressed in the UR analysis. Again requirements are identified in the two priorities as defined and used in section 2.

3.2 Software Requirements

Table 5: Requirements for Fast Physical Shielding Analysis Model.

SR ID	Description of Requirement	Importance	References & Notes
SR 1	IRONSSIS shall treat HI propagation in 1D planar shields NOTE: To treat spherical shields, it has been agreed to that the ESHIEM Project attempt to use an implementation of Seltzer's algorithm developed by RadMod Research as part of the IRPAM Project, and use this as a post-process to IRONSSIS/MULASSIS.	1	UR 31
SR 2	IRONSSIS shall perform a fast shielding analysis based on approximate particle transport, rather than precise Monte Carlo transport	1	UR 33
SR 3	IRONSSIS shall be relevant to all naturally occurring ions treated by Geant4, $1 \leq Z \leq 92$, and normal or cosine-law incidence.	1	
SR 4	IRONSSIS shall be relevant to all target materials treated by Geant4 NOTE: It seems to be reasonably possible to generate shielding information (and a tool based on this) which is not specific to just a few materials.		UR 34.1, 34.2
SR 5	The IRONSSIS shall estimate as a function of depth and particle species: (SR 5.1) HI fluxes or fluences (SR 5.2) total ionising dose (SR 5.3) total non-ionising dose in Si and GaAs (SR 5.4) total non-ionising dose in materials based on NIEL coefficients provided by user. NOTE 1: Following the UR definition, the Agency also expressed an interest in the user being able to provide their own coefficients for other materials, and so this is specified as a separate requirement. NOTE 2: The generation of LET spectra will be the responsibility of the interface code (PHP scripts) which process the flux calculated for each of the shield outputs (UR 35.4)	1 1 2 2	UR 35

SR ID	Description of Requirement	Importance	References & Notes
SR 6	IRONSSIS have a method to propagate the uncertainties in the calculation process. NOTE 1: Estimation of the errors in the physical shielding calculation is much more challenging, and based on the accuracy of the stopping power information and approximations resulting from ignoring the other effects, particularly multiple scattering. This is best resolved by comparison with Monte Carlo results for a few cases. And Monte Carlo results themselves have systematic and statistical errors - the former are again challenging to estimate and highly dependent upon the physical conditions (materials, particles, energies, thickness) simulated. NOTE 2: From the perspective of the accuracy of the shielding analysis, the largest uncertainty arises from assuming that shielding analysis can be based on results from 1D shielding conditions.	1	UR 36
SR 7	The user shall be able to provide modified estimates of the errors in the model, which can then be propagated.	1	UR 37

In addition to the above IRONSSIS-specific user requirements, the following are also applicable.

Table 6: Interface requirements.

UR ID	Description of Requirement	Importance	References & Notes
SR 8	The interfaces shall be consistent with those already employed for: (SR 8.1) SEPEM (SR 8.2) SPENVIS, *i.e.* output files shall conform to SPENVIS .CSV file format to enable easy interfacing with SPENVIS and data visualisation	1	UR 41
SR 9	IRONSSIS shall be able to treat very thin (~0.01 mm) shielding conditions.	1	UR 42
SR 10	IRONSSIS will have an interface similar to that used in MULASSIS, and wherever possible use identical commands and command structure.	1	UR 41
SR 11	There shall be an Interface Control Document/Specification.	1	

Table 7: Operational environment and maintenance.

UR ID	Description of Requirement	Importance	References & Notes
SR 12	IRONSSIS will be consistent with the design approach and software practices adopted for the existing SEPEM system.	1	UR 44

UR ID	Description of Requirement	Importance	References & Notes
SR 13	IRONSSIS will be based on Geant4/MULASSIS and shall be consistent with: (SR 13.1) The existing Geant4 code in terms of modularisation and coding practice (SR 13.2) The design and coding practices of MULASSIS	 1 1	UR 45
SR 14	The software shall be maintained for one year following acceptance by ESA.	1	UR 46

Table 8: Verification and validation requirements.

UR ID	Description of Requirement	Importance	References & Notes
SR 15	Verification and validation shall take place in accordance with ECSS-E-40A.	1	UR 47
SR 16	Validation of environment models shall include comparisons with results from other effects tools, particularly MULASSIS.	1	UR 48

Table 9: Performance requirements.

UR ID	Description of Requirement	Importance	References & Notes
SR 17	The performance of IRONSSIS shall be such as to allow the simulation of the shield in less than 1 minute.	1	

3.3 Traceability

In the following requirements traceability matrices

- UR 32 is not covered, since the requirement is met by the current version of SEPEM using MULASSIS, and we are only addressing IRONSSIS requirements here.
- UR 35.4 will be met by the existing PHP scripts used with SEPEM, not from IRONSSIS.
- SR 5.4 is an amplified requirement requested by the Agency.
- SR 11 and SR 17 were not specifically addressed within the original User Requirements

	SR 1	SR 2	SR 3	SR 4	SR 5.1	SR 5.2	SR 5.3	SR 5.4	SR 6	SR 7	SR 8.1	SR 8.2	SR 9	SR 10	SR 11
UR 31	X														
UR 32															
UR 33		X													
UR 34.1				X											
UR 34.2				X											
UR 35.1					X										
UR 35.2						X									
UR 35.3							X								
UR 35.4															
UR 36									X						
UR 37										X					
UR 41.1											X			X	
UR 41.2												X		X	
UR 42													X		
UR 44															
UR 45															
UR 46															
UR 47															
UR 48.3															

Table 10: User requirements / software requirements cross-reference matrix (part 1 of 2).

	SR 12	SR 13.1	SR 13.2	SR 14	SR 15	SR 16	SR 17
UR 31							
UR 32							
UR 33							
UR 34.1							
UR 34.2							
UR 35.1							
UR 35.2							
UR 35.3							
UR 35.4							
UR 36							
UR 37							
UR 41.1							
UR 41.2							
UR 42							
UR 44	X						
UR 45		X	X				
UR 46				X			
UR 47					X		
UR 48.3						X	

Table 11: User requirements / software requirements cross-reference matrix (part 2 of 2).

4 Shielding Analysis Algorithm

4.1 Basis of algorithm

Within this chapter, we briefly discuss the physical and mathematical basis for a fast shielding analysis algorithm relevant to ions. The complexity of modern Monte Carlo techniques for performing such calculations is driven largely by the inclusion of particle tracking in very fine detail, particularly treating multiple scattering from nuclei in the target materials and the generation and propagation of secondaries from nuclear interactions. In order to perform a more rapid shielding calculation, the following simplifications should be made:

- We shall treat only 1D slab shields comprising single and multiple layer planar shields materials.
- The calculation should assume a straight-ahead, continuous slowing-down approximation (CSDA) for the ion propagation.
- The angle of incidence for the ions on the shield is considered either normal or cosine-law (the latter to represent an isotropic radiation source incident upon a finite plane).
- Range-versus-energy information for each ion will be based on an interpolation (*e.g.* cubic splines) of range-energy data at discrete ion energies, determined from integrating Geant4 ion stopping power model results. In particular, Geant4 includes stopping power models based on ICRU-73 [15] and the PASS model of Sigmund and Schinner [16][17] generated as part of ESAs Rapid Reverse Monte Carlo and Ion Physics contract [18].
- Attenuation of the incident ions by nuclear interactions of ions will be treated based on the total non-elastic nuclear-nuclear cross-section modules within Geant4 to generate the necessary cross-section data to account for attenuation.
- Within the scope of the resources available it is not possible to treat the effects of multiple scattering or the generation of secondaries from nuclear interactions.
- The principal output from the model should be:
  - Residual primary ion flux or fluence.
  - From the flux/fluence, TID (potentially integrated over ion species) can be estimated within the model using electronic stopping power data calculated using Geant4 physics. Alternatively, to be consistent with the existing MULASSIS code, the TID can be calculated from the difference in the incident and exiting energy flux, and the mass of the layer.
  - For TNID, non-ionising energy loss (NIEL) coefficients already existing for protons in a number of different semiconductor materials and are implemented in MULASSIS. The fast 1D shielding model will therefore allow TNID estimates to be generated for protons in these materials, or for other ions and materials where the user is able to provide suitable NIEL coefficients.
  - LET spectra, integrated over ion species, can also be generated and used subsequently with existing SEE-rate prediction tools in SEPEM.

4.2 Equation for shielded ion spectrum

When dealing with a CSDA effect, we need to treat the phase-space change of the incident ion spectrum. Consider a spectrum $\psi_i(E_i,\theta)$ to be the differential spectrum of ions incident upon a shield of thickness T at angle θ. For each incident energy per nucleon, En_i, the energy of the particle after the shield is En_f where:

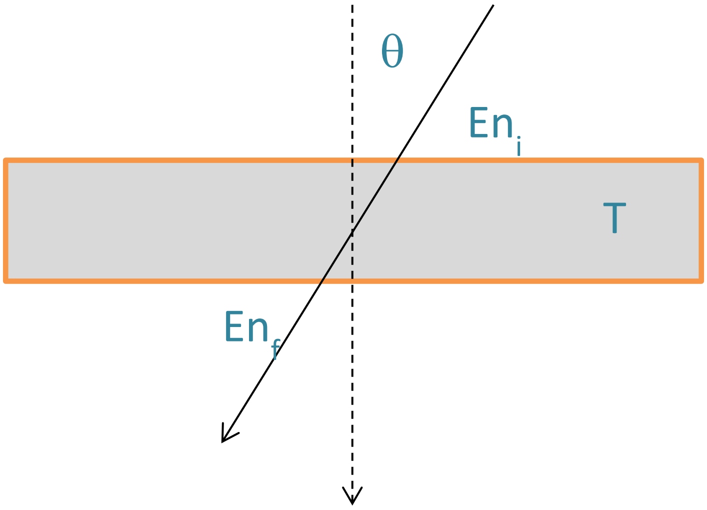


Figure 1: Simple slab shielding geometry.

$$\frac{T}{cos\theta} = R(En_i) - R(En_f)$$

Equation 1

where R(En) is the range of the ion in the material for energy per nucleon En, given by the integral of the reciprocal stopping power:

$$R(En) = \int_0^{En} \frac{A_P}{S(En')} dEn'$$

Equation 2

where A_P is the nucleon number of the projectile Therefore, rearranging Equation 1 gives:

$$En_f = R^{-1}\left[R(En_i) - \frac{T}{\cos\theta}\right]$$

or

$$En_i = R^{-1}\left[R(En_f) + \frac{T}{\cos\theta}\right]$$

Equation 3

Here, R^{-1} is the inverse range equation. The equation for the spectrum of ions leaving the geometry at angle θ therefore becomes:

$$\psi_f(En_f, \theta) = \psi_i(En_i, \theta)\frac{S(En_i)}{S(En_f)}$$

or

$$\psi_f(En_f,\theta) = \psi_i\left(R^{-1}\left[R(En_f)+\frac{T}{\cos\theta}\right],\theta\right)\frac{S\left(R^{-1}\left[R(En_f)+\frac{T}{\cos\theta}\right]\right)}{S(En_f)}$$

Equation 4

Obviously for normal incidence this reduces to:

$$\psi_{\perp f}(En_f) = \psi_{\perp i}\left(R^{-1}\left[R(En_f)+T\right]\right)\frac{S\left(R^{-1}\left[R(En_f)+T\right]\right)}{S(En_f)}$$

Equation 5

whilst for isotropic incidence, this becomes:

$$\psi_{\cap f}(En_f) = \int_0^{\pi/2} \psi_{\cap i}\left(R^{-1}\left[R(En_f)+\frac{T}{\cos\theta}\right],\theta\right)\frac{S\left(R^{-1}\left[R(En_f)+\frac{T}{\cos\theta}\right]\right)}{S(En_f)} sin\theta d\theta$$

Equation 6

When dealing with more than one material layer, it relatively easy to apply these conversions to the spectra at each intervening layer.

From the perspective of uncertainties in the intensity of the incident spectrum, $\delta_I(En_i,\theta)$, these can be propagated to form uncertainties in the emitted spectrum:

$$\delta_f(En_f,\theta) = \delta_i\left(R^{-1}\left[R(En_f)+\frac{T}{\cos\theta}\right],\theta\right)\frac{S\left(R^{-1}\left[R(En_f)+\frac{T}{\cos\theta}\right]\right)}{S(En_f)}$$

Equation 7

as we assume here that the error in the spectrum dominates over potential errors in the stopping power and range equations, and that the error in the material thickness can be neglected.

4.3 Attenuation by nuclear interactions

Losses of ions by nuclear interactions can be treated using existing formulae for the total inelastic cross-section, such as the general formula from Tripathi *et al*, or a separate formula from Tripathi *et al* which is intended to be specifically for light ion/target conditions [19][20]. Once the microscopic cross-section, σ_{TH}, is defined for each ion and target material, then an attenuation exponent can be generated. For an element, this is given by the equation:

$$\lambda = \frac{N_A A_P}{W_T}\int_{E_i}^{E_f}\frac{\sigma_{TH}(En')\times 10^{-27}}{S(En')}dEn'$$

Equation 8

where N_A is the Avogadro constant (mol^{-1}), W_T is the atomic weight of the target (g/mol), and σ_{TH} is given in millibarns, and S is in $MeVcm^2/g$.

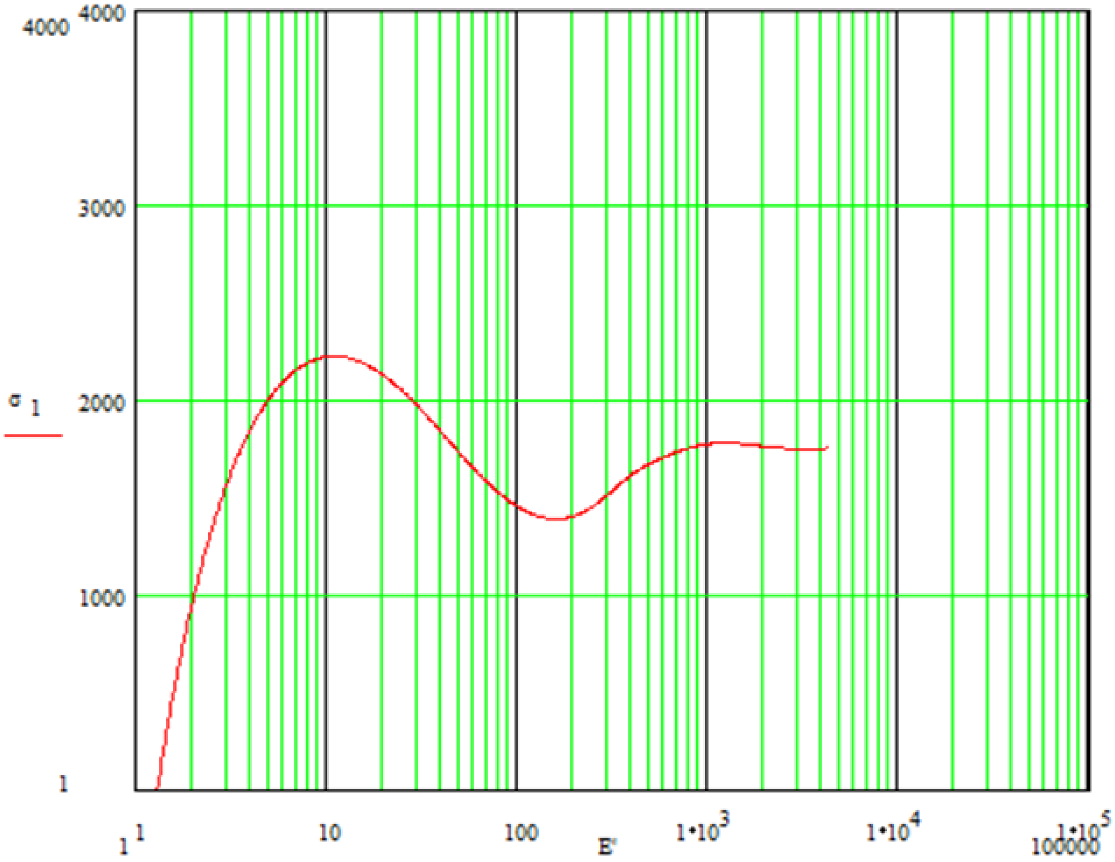


Figure 2: Example total inelastic cross-section curve for ^{23}Na ions on Al, based on standard Tripathi algorithm (σ in millibarns, and E in MeV/nuc).

The intensity of the ion spectrum following passage through the slab shield is therefore based on the modified version of Equation 4 with λ defined in Equation 8:

$$\psi_f\left(En_f,\theta\right)=e^{-\lambda}\psi_i\left(R^{-1}\left[R\left(En_f\right)+\frac{T}{\cos\theta}\right],\theta\right)\frac{S\left(R^{-1}\left[R\left(En_f\right)+\frac{T}{\cos\theta}\right]\right)}{S\left(En_f\right)}$$

Equation 9

If we assume that the uncertainties within the intensity of the spectrum are much larger than the other contributors, then the uncertainties in the final spectrum are $\delta_f(En_f,\theta)$, and dependent on the errors in the incident spectrum based on:

$$\delta_f\left(En_f,\theta\right)=e^{-\lambda}\delta_i\left(R^{-1}\left[R\left(En_f\right)+\frac{T}{\cos\theta}\right],\theta\right)\frac{S\left(R^{-1}\left[R\left(En_f\right)+\frac{T}{\cos\theta}\right]\right)}{S\left(En_f\right)}$$

However, it is acknowledged that compared to the errors in the spectrum, the uncertainty in σ_{TH} may still be important, albeit difficult to determine without extensive nuclear irradiation results.

5 Interface Control Specification

The default IRONSSIS macro commands will be based on MULASSIS-type commands in order to maintain insofar as possible a similar interface between SEPEM (PHP scripts) and MULASSIS, and since it has been decided that IRONSSIS will also perform MULASSIS Monte Carlo simulations as well as the non-Monte-Carlo (non-MC) simulations described in section 4.

Table 12 to Table 18 summarise the Geant4 commands to control geometry definition, source particle environment, physical models applied, type of analysis performed, numerical integration controls and execution parameters. Commands in the first column which are in blue are traditional MULASSIS commands, whilst those which are in brown are new IRONSSIS commands. Column 3 text which is in red highlights where command input options are restricted in order to perform an IRONSSIS calculation.

When operated in non-MC mode, IRONSSIS will analyse only the primary particles, whilst in MC mode (traditional MULASSIS) all particle may be treated subject to the physics conditions.

Command	Arguments and argument restrictions	Command function and restrictions
`/geometry/material/add <name> <material> <density>`	G4String name, material; G4double density>0.0	Creates a material with material name name with the chemical formula material and of density density (g/cm^3).
`/geometry/material/addNIST <material>`	G4String material	Add a NIST material by name
`/geometry/material/listNIST <material>`	G4String type	List NIST material by type = “all”, “simple”, “compound” or “hep”
`/geometry/material/delete <name>`	G4String name	Deletes the material called name.
`/geometry/material/list`		Lists the currently defined materials and the associated index for the material.
`/geometry/layer/shape <shape>`	G4double shape = “slab” or “sphere”	Defines the geometry type for the multi-layer shield as planar or spherical. ***In order to execute IRONSSIS, the geometry shape should be slab, whilst MULASSIS run execute for either slab or sphere geometries.***
`/geometry/layer/add <layerIndex> <materialName> <colourIndex> <thickness> <unit>`	G4int layerIndex; G4String materialName; G4int colourIndex, G4double thickness>0.0; G4String unit = "mum", "mm", "cm", "m", "km", "mg/cm2", "g/cm2", "kg/m2	Adds a shield layer after layer number layerIndex comprising material materialName of thickness thickness (in units unit) and associated colour defined by colourIndex
`/geometry/delete/layer <layerIndex>`	G4int layerIndex≥0	Deletes the layer of index layerIndex in the shield layer list. If layerIndex==0, all layers are deleted.
`/geometry/layer/list <layerIndex>`	G4int layerIndex≥0	List the thickness of layer layerIndex, associated material and colour attributes.
`/geometry/colour/add <colourName> <R> <G> <B>`	G4String colourName; G4double 0≤R≤1, 0≤G≤1 , 0≤B≤1	Defines a colour according to its RGB intensity.
`/geometry/colour/list`		Lists the available colour palette and the associated colourIndex.

/geometry/default		Sets the shield thicknesses to the SHIELDOSE2 defaults
/geometry/update		Updates the geometry information after changes requested by the user. This is a mandatory command after any changes made to the geometry.

Table 12: Summary of IRONSSIS geometry commands.

Command	Arguments and argument restrictions	Command function and restrictions
/gps/particle <particleType>	G4String particleType	Sets the incident particle type. ***In order to execute IRONSSIS, the particle type should be an ion, whilst other options are suitable for a MULASSIS run.***
/gps/ion <Z> <A> <Q> <E>	G4int Z>0, A>0 Q; G4double E≥0.0	For an ion source particle, sets the Z, A and excitation energy of the nucleus and charge (Q in integer units of e) for the ion.
/gps/energy <energy> <unit>	G4double energy; G4String unit	Sets the source particle distribution to monoenergetic of energy energy with units of energy unit.
/gps/energytype <energyDis>	G4String energyDis = “Mono”, “Lin”, “Pow”, “Exp”, “User”, “Arb”	Sets the source particle energy distribution to: Mono - monoenergetic Lin - linear Pow - power-law Exp - exponential User - user-defined histogram Arb - arbitrary point-wise spectrum
/gps/ene/min <eMin> <unit>	G4double eMin>0.0; G4String unit	Sets the minimum for the energy distribution.
/gps/ene/max <eMax> <unit>	G4double eMax=0.0; G4String unit	Sets the maximum for the energy distribution.
/gps/ene/alpha <alpha>	G4double alpha	Sets the α for the power-law distribution of the form y=AEα where A is a constant.
/gps/ene/gradient <gradient>	G4double gradient	Sets the gradient for linear distributions
/gps/ene/intercept <intercept>	G4double intercept	Sets the intercept for the linear distribution
/gps/hist/type	G4String type	Sets the type of histogram to be defined
/gps/hist/reset		Resets the histogram data.
/gps/hist/point <eHi> <weight>	G4double eHi>0.0, weight≥0.0	Allows definition of histogram, point-by-point.
/gps/hist/inter <interpType>	G4String interpType	Sets the interpolation type for point-wise data: Lin - linear Log - power-law interpolation Exp - exponential Spline - cubic spline

Command	Arguments and argument restrictions	Command function and restrictions
/gps/ang/type <angDis>	G4String angDis = “planar”, “cos”	Sets the angular distribution type planar – unidirectional beam cos - cosine-law ***In order to execute IRONSSIS, the angular distribution should be either of these two option, whilst other options are suitable for a MULASSIS run.***
/gps/direction <x> <y> <z>	G4double x, y, z; z>0.0	Sets direction cosines of incident beam (planar) particle source.
/gps/ang/mintheta <minT> <unit>	G4double minT > 0.0 && < 89.9 deg G4String unit = “deg”, “rad”	Minimum angle of incidence for incident cosine-law particle source
/gps/ang/maxtheta <maxT> <unit>	G4double maxT > minT G4String unit = “deg”, “rad”	Maximum angle of incidence for incident cosine-law particle source

Table 13: Summary of IRONSSIS gps commands.

Command	Arguments and argument restrictions	Command function and restrictions
/phys/scenario <state>	G4String state = “hadron-leem-ln”, “hadron-em-ln”, “hadron-em_pen”, “ln hadron-em_liv-ln", "hadron-leem+ln”, “hadron-em+ln”, “hadron-em_pen+ln”, “hadron-em_liv+ln", "hadron+leem-ln”, “hadron+em-ln”, “hadron+em_pen-ln”, “hadron+em_liv-ln" "hadron+leem+ln”, “hadron+em+ln”, “hadron+em_pen+ln”, “hadron+em_liv+ln”, “em”, “leem”, “em_liv”, “em_pen”, “em_opt3”	Switches between different Geant4 physics options. It is recommended to use “em_opt3” of only EM transport for ions is required, and “hadron-leem-ln” or “hadron+leem-ln” if nucleus-nucleus collisions are to be treated.

Table 14: Summary of IRONSSIS physics commands.

Command	Arguments and argument restrictions	Command function and restrictions
/analysis/normalise <fluence> <unit>	G4double fluence; G4Stringunit =”cm2”, “m2”	Sets the incident particle fluence to fluence in units of unit.
/analysis/file <filename>	G4String filename	Sets the name of the CSV RPT files.

Command	Arguments and argument restrictions	Command function and restrictions
/analysis/fluence/add <layerIndex>	G4int layerIndex≥0	Selects the layer denoted by layerIndex for which particle fluence analysis must be performed.
/analysis/fluence/delete <layerIndex>	G4int layerIndex	Deselects layer layerIndex for particle fluence analysis.
/analysis/fluence/list		Identifies the layers for which particle fluence analysis is to be performed
/analysis/fluence/unit unit	G4String unit = "cm2", "m2"	Sets the units of fluence used in the output.
/analysis/fluence/type type	G4String type = "omni", "planar"	Selects whether the standard (omnidirectional) fluence or planar fluence is measured. ***In order to execute IRONSSIS correctly, the omni fluence type should be selected, whilst other options are suitable for a MULASSIS simulation.***
/analysis/fluence/energy /mode <mode>	G4String <mode> = "LINEAR", "LIN", "linear", "lin", "LOGARITHMIC", "LOG", "logarithmic", "log", "ARBITRARY", "ARB", "arbitrary", "arb"	Sets the energy binning-scheme mode for particle fluence data.
/analysis/fluence/energy /default		Applies the default energy binning-scheme (see Appendix B of [9]).
/analysis/fluence/energy /min <eMin> <unit>	G4double eMin ≥ 0.0 && eMin ≤ eMax; G4String unit = "eV", "keV", "MeV", "GeV", "TeV", "PeV"	Sets the lower-edge of the lowest bin in the energy binning-scheme.
/analysis/fluence/energy /max <eMax> <unit>	G4double eMax ≥ eMin; G4String unit = "eV", "keV", "MeV", "GeV", "TeV", "PeV"	Sets the upper-edge of the highest bin in the energy binning-scheme.
/analysis/fluence/energy /nbin <Nebin>	G4int N >0	Sets the number of bins in the energy binning-scheme. Nebin is forced to unity if eMax==eMin This command is not available for arbitrary binning schemes.
/analysis/fluence/energy /list		Lists the current energy binning-scheme.
/analysis/fluence/energy /add <eEdge> <unit>	G4double eEdge; eMin<eEdge<eMax; G4String unit = "eV", "keV", "MeV", "GeV", "TeV", "PeV"	Adds a bin-edge. Only available for arbitrary binning-schemes.
/analysis/fluence/energy /delete <eEdge> <unit>	G4double eEdge; G4String unit = "eV", "keV", "MeV", "GeV", "TeV", "PeV"	Deletes and exsiting bin-edge. Only available for arbitrary binning schemes.
/analysis/fluence/energy /clear		Delete all bin-edges between eMin and eMax (Nebin set to 1).
/analysis/dose/add <layerIndex>	G4int layerIndex≥0	Selects the layer denoted by layerIndex for which Edep/TID analysis must be performed.
/analysis/dose/delete <layerIndex>	G4int layerIndex	Deselects layer layerIndex for Edep/TID analysis.
/analysis/dose/list		Identifies the layers for which Edep/TID analysis is to be performed

Command	Arguments and argument restrictions	Command function and restrictions
/analysis/dose/unit <unit>	G4String unit = "eV", "keV", "MeV", "GeV", "TeV", "PeV", "rad", "Gy"	Selects the units by which dose or energy deposition in the layers shall be measured
/analysis/niel/add <layerIndex>	G4int layerIndex≥0	Selects the layer denoted by layerIndex for which NIEL analysis must be performed.
/analysis/niel/delete <layerIndex>	G4int layerIndex	Deselects layer layerIndex for NIEL analysis.
/analysis/niel/list		Identifies the layers for which NIEL analysis is to be performed
/analysis/niel/function <NIELCoeffs>	G4String NIELCoeffs = "cern", "CERN", "jpl" or "JPL"	Selects the set of NIEL coefficients denoted by NIELCoeffs for use with particle fluence data.

Table 15: Summary of IRONSSIS analysis commands.

Command	Arguments and argument restrictions	Command function and restrictions
/control/execute <macrofile>	G4String macrofile = "display1.mac", "display2.mac", "display3.mac", "display4.mac", "display5.mac", or "display6.mac"	Selection of visualisation options, by appropriate choice of macrofile.
/ironssis/beamOn		Perform shielding analysis, equivalent to /run/beamOn for MULASSIS simulation.
/ironssis/error/setGlobalPercent <errorPC>	G4double errorPC >= 0.0 && <= 1.0E+10	Sets the percentage error to be applied to the IRONSSIS results. Default is 0.0
/ironssis/error/getGlobalPercent		Displays the percentage error to apply to the results.

Table 16: Summary of other miscellaneous commands of IRONSSIS.

Command	Arguments and argument restrictions	Command function and restrictions
/ironssis/integration/setMaxTIntv <maxTIntv> <unit>	G4double maxTIntv >= 0.01 deg && <=5.0 deg G4String unit = “deg” or “rad”	Defines the maximum interval in theta for integration of incident (cosine-law) flux. Default is 0.2 degrees.
/ironssis/integration/setMaxTLimit <maxTLimit> <unit>	G4double maxTIntv >=85.0 deg && <= 89.999 deg G4String unit = “deg” or “rad”	Defines the maximum angle to the normal up to which the incident (cosine-law) flux will be integrated. Default is 89.9 degrees
/ironssis/integration/setNMinTIntv <minTIntv>	G4int minTInv >= 1 && <= 100	Set the minimum number of angle intervals for Simpson 3/8 integration within each fluence bin. Default is 20.
/ironssis/integration/minEpn <minEpn> <unit>	G4double minEpn >= 0.01 MeV && <= 10.0 GeV G4String unit = “eV”, “keV”, “MeV”, “GeV”	Defines the minimum energy per nucleon from which the numerical integration of the incident spectrum will be performed. Default is 0.01 MeV
/ironssis/integration/nEpd <nEpd>	G4int nEpd >= 1 && <= 100	Number of energies per decade over which the spectra are defined. Default is 20.

Command	Arguments and argument restrictions	Command function and restrictions
`/ironssis/integration/nEpts <nEpts>`	G4int nEpts >= 22	Total number of energies to be used for numerical integration of the incident spectrum. Default is 140, therefore maximum energy is 10 GeV/nuc, if minEpd and nEpd are used with default values.
`/ironssis/integration/nEint <nEint>`	G4int nEint >= 4 && <= 100	The attenuation cross-section is integrated over for each material and over each of the nEpd intervals per decade in energy using a Simpson 3/8 integration. nEint defines the number of intervals over which the energy integration is performed. Default is 22.
`/ironssis/integration/listParameters`		Lists all the current parameter values for integration process.

Table 17: Summary of commands to control numerical integration. These parameters should not be changed unless the user has a clear understanding of the consequences on the numerical integration processes.

Command	Arguments and argument restrictions	Command function and restrictions
`/ironssis/niel/name <name>`	G4String name	Allows user to define a name/designation for the new NIEL coefficients
`/ironssis/niel/datapoint <userEnergy> <userNIEL>`	G4double userEnergy > 0.0, userNIEL >= 0.0 G4String unit = “deg” or “rad”	Defines the maximum angle to the normal up to which the incident (cosine-law) flux will be integrated. Default is 89.9 degrees
`/ironssis/niel/clear`		Clears all NIEL curve data-points already entered for the user-defined NIEL curve.
`/ironssis/niel/list`		Lists the table of data-points for the user-defined NIEL curve.

Table 18: Summary of commands to control inputting of new NIEL coefficients.

NOTE 1:

In order to perform simulations for different ion species, it is assumed that the IRONSSIS runs will be repeated by SEPEM, or a sufficient Geant4 macro file will be defined covering each of the ion species for which shielding information is required.

NOTE 2:

In order to calculate LET spectra, the existing PHP scripts in SEPEM which are already used with MULASSIS can be reused with IRONSSIS results with a few modifications. Indeed, the IRONSSIS application will operate as a MULASSIS Monte Carlo simulation as well, if default MULASSIS commands are used.

NOTE 3:

An IRONSSIS simulation cannot be executed if one or more of the following conditions exist:

- The geometry type is set to SPHERE (simulations can only be performed for SLAB geometry)[1].
- The incident particle is not an ion.
- The angular distribution for the incident particles is not planar and is not cosine-law.
- The incident particles have no component which is from the +Z direction (*i.e.* they are incident upon only the rear of the shield not from the front).
- If ion fluence analysis is performed and the fluence analysis type is PLANAR rather than OMNI-DIRECTIONAL.

[1] NOTE: To treat spherical shields, it has been agreed to that the ESHIEM Project attempt to use an implementation of Seltzer’s algorithm developed by RadMod Research as part of the IRPAM Project, and use this as a post-process to IRONSSIS/MULASSIS.

6 Component Description

6.1 Introduction

The IRONSSIS application is in fact a heavily modified (branch of) the MULASSIS application incorporating 26 new files defining 15 new C++ classes to provide the additional non-MC functionality. In addition, modifications have been made to some of the existing MULASSIS classes, *e.g.* to allow the MULASSIS detector analysis classes (such as MLFluenceAnalyser) to be a base class to the new classes, or to extend tallying histograms to treat MC and non-MC data depending upon the type of simulation.

This release of IRONSSIS is based on version 1.25 of MULASSIS, which uses Geant4 v9.6 patch02.

The names and locations of the C++ source code file for the main program and executable remain unchanged – “trunk/mulassis.cc” and “build/mulassis”. It is hoped that in due course and after discussion with ESA and SpaceIT/PSI (which is currently maintaining MULASSIS), IRONSSIS will formally be:

- Integrated back into the SVN “trunk” of MULASSIS and the publicly distributed MULASSIS will have all the MC and non-MC functionality described here, or
- Remain largely as a “branch”, but with updates in the base MULASSIS “trunk” regularly propagated into the “branch”.

Indeed significant effort has been given to designing and then redesigning the object-oriented IRONSSIS in a manner that is consistent with the current software architecture of MULASSIS, and so that the IRONSSIS functionality can be added relatively easily to a future version of (MC-only) MULASSIS.

6.2 New IRONSSIS (IR*) classes

The table below summarises the new classes for IRONSSIS. For more information at detailed-design level, the reader is referred to the comments within the source code for these files, and also the C++ header files.

New class	Base class	Brief description of class
IRRunManager	MLRunManager	This is the overall run manager for the Geant4 application, with much of its functionality derived from MLRunManager. Other than performing the normal functions for MULASSIS, this class has a little additional code to introduce the run-messenger (which allows the additional IRONSSIS commands to execute the IRONSSIS calculation and set verbose level) and to invoke the relevant analysis manager member function to start the IRONSSIS calculation.
IRRunMessenger	G4UImessenger	Class to add additional commands to execute an IRONSSIS simulation. Note that all other commands for MULASSIS should be available through the usual MULASSIS messenger classes. (SR 10)
IRAnalysisManager	MLAnalysisManager	The IRAnalysisManager class is similar to the MLAnalysisManager class and is the main class which adds the IRONSSIS functions to MULASSIS. It includes the IRFluenceAnalyser, IRDoseAnalyser and IRNIELAnalyser objects which accumulate/output the analytical results from the IRONSSIS analysis and based on the source conditions, establishes the IRLayerFluxDose objects for each layer which define the change in spectrum between the input and output layers. NOTE: Like MLAnalysisManager, there must only be one instance of IRAnalysisManager. (SR 1, SR 2, SR 3, SR 4, SR 5, SR 6)
IRAnalysisMessenger	G4UImessenger	This is a G4 messenger class associated with IRAnalysisManager. For the moment the class enables the user to manipulate some of the parameters used for the integration within IRONSSIS, definition of the percentage error to propagate into the results, and definition of user-supplied NIEL coefficients as a function of energy to apply to ions. (SR 1, SR 2, SR 3, SR 4, SR 5, SR 6, SR 7, SR 10)
IRFluenceAnalyser	MLFluenceAnalyser	IRFluenceAnalyser is a derived version of MLFluenceAnalyser and includes additional or modified functions to set and output the fluence values for each layer based on the IRONSSIS calculation using the IRLayerFluxDose objects for each layer. This is not a singleton, but there should be only one instance of this object, controlled by the singleton IRAnalysisManager which instantiates IRFluenceAnalyser. (SR 5.1, SR 10)
IRDoseAnalyser	MLDoseAnalyser	IRDoseAnalyser is a derived version of MLDoseAnalyser and includes additional or modified functions to set and output the dose values for each layer based on the IRONSSIS calculation using the IRLayerFluxDose objects for each layer. This is not a singleton, but there should be only one instance of this object, controlled by the singleton IRAnalysisManager which instantiates IRDoseAnalyser. (SR 5.2, SR 10)

New class	Base class	Brief description of class
IRNIELAnalyser	MLNIELAnalyser	IRNIELAnalyser is a derived version of MLNIELAnalyser and includes additional or modified functions to set and output the dose values for each layer based on the IRONSSIS calculation using the IRLayerFluxDose objects for each layer. This is not a singleton, but there should be only one instance of this object, controlled by the singleton IRAnalysisManager which instantiates IRNIELAnalyser. (SR 5.3, 5.4, SR 10)
IRLayerFluxDose	(none)	Classed used by the analysis/detector classes to quantify the propagation of ions through a single layer of the geometry. Given range and stopping-power information as a function of energy, the instantiated objects of this class calculates the spectrum of ions leaving the shield from the spectrum entering as a function of incidence angle. The spectra can be integrated using relevant member functions of this class (invoking member functions of the IRDiffSpectrumInterpolation class) to give fluence over defined energy ranges, as well as TID and TNID. (SR 1, SR 2, SR 3, SR 4, SR 5, SR 9)
IRDEDXInterpolation IRRangeInterpolation IRInvRangeInterpolation	SplineInterpolation	This contains several classes to perform an interpolation of range-versus energy data and stopping-power-versus-energy data. Each class is derived from a spline interpolation class, and the analysis is performed as a spline to the data or log data: IRDEDXInterpolation lin-DEDX as a function of log-E IRRangeInterpolation log-Range as a function of log-E IRInvRangeInterpolation log-E as a function of log-Range The extrapolation beyond the endpoints are assumed to be linear in the log(y) or lin(y) versus log(x) coordinates (depending upon the function as described above). (SR 1, SR 2, SR 3, SR 4, SR 9)
IRLambdaInterpolation	SplineInterpolation	Uses the total inelastic cross-section information for nuclear-nuclear collisions to determine the attenuation of particles through materials as a function of energy. (SR 1, SR 2, SR 3, SR 4, SR 5)
IRDiffSpectrumInterpolation	SplineInterpolation	IRDiffSpectrumInterpolation represents a differential spectrum as a series of points and the interpolation (cubic spline in log differential fluence versus log energy). It includes member functions to perform integration of fluence over energy, integration of energy fluence for ions leaving the layer, and integration of NIEL over differential spectrum to determine TNID. NOTE: the spectrum has power-law extrapolation at end-points (*i.e.* linear in log-log-space). (SR 1, SR 2, SR 3, SR 4, SR 9)
IRPhysicsUtil	MLPhysicsUtil	The IRPhysicsUtil class adds functions to determine the nuclear inelastic cross section for a material. This is then used within the IRLambdaInterpolation object in the same manner that stopping power and range information from MLPhysicsUtil is used by IRDEDXInterpolation, IRRangeInterpolation and IRInvRangeInterpolation. (SR 1, SR 2, SR 3, SR 4)
SplineInterpolation		This is a class to define a spline interpolation based on an array or std::vector input for x and y. This class also includes member functions to perform trapezoid integration and Romberg integration.

Table 19: Summary of new IRONSSIS classes.

6.3 Modifications to MULASSIS (ML*) classes

The principal changes to the existing MULASSIS classes include:

- Changes to MLAnalysisManager, MLFluenceAnalyser, MLDoseAnalyser, MLNIELAnalyser to allow access to variables by derived classes (*i.e.* modified variables from "private" to "protected").
- Histo1DVar modified so that bins contents of histograms can be changed directly without repeated tallying.
- MLNIELFunction has additional member functions to allow a user-defined NIEL function to be used instead of the existing values in MLNIELData. Note that this feature is currently only available for IRONSSIS non-MC calculations, but could be extended to conventional MULASSIS MC calculations.
- MLVersion modified to refer to IRONSSIS (note, this may be a temporary measure before integrating IRONSSIS functionality back into the "trunk" of MULASSIS).

6.4 File locations

The principal subdirectories within the main ironssis directory are:

- ironssis/trunk: contains the mulassis.cc source code and build information.
- ironssis/trunk/src: all C++ source code.
- ironssis/trunk/include: all C++ header files.
- ironssis/trunk/macro: standard MULASSIS Geant4 macros for display drivers.
- ironssis/trunk/test: test example files, which can be used for both conventional MC and IRONSSIS calculations.

The examples within the test subdirectory are:

- ML_23Na_Al: Al-23 ions incident on a single layer of Al.
- ML_23Na_Al+C: Al-23 ions incident on a single layer of Al followed by a layer of C.
- ML_23Na_Al: Al-23 ions incident on a single layer of Al, with treatment of attenuation by nucleus-nucleus collisions.
- ML_23Na_Al+C: Al-23 ions incident on a single layer of Al followed by a layer of C, with treatment of attenuation by nucleus-nucleus collisions.
- ML_23Na_Al+NIEL: Al-23 on a single layer of Al, with use of user-defined NIEL function.
- ML-23Na_LoDens: A series of examples in which the shield layers comprise vacuum in order to test fluence calculation.
- ML_p_Al: protons incident on a single layer of Al.
- ML_p_Al+NIEL: protons incident on a single layer of Al, with use of user-defined NIEL function.

7 Compilation, Build and Execution

7.1 Compilation/build

The directory structure and file locations are described within section 6.4. It is assumed that Geant4 v9.6 patch02 is implemented on the local system and the associated environment variables defined (see http://geant4.cern.ch/support/download.shtml). The commands to compile and build the IRONSSIS software are:

```
cd ironssis/build
cmake ../trunk
make
```

The compilation/build tests have been performed with gcc/g++ compiler version 4.6.2 under OpenSUSE Linux v12.1.

7.2 Execution

To execute the application with a macrofile named “theMacroFile.gmac” in directory ironssis:

```
build/mulassis theMacroFile.gmac
```

or to run one of the tests in ironssis/trunk/test (such as ML_23Na_Al/23Na_-1.5_100-10000MeV_90.gmac):

```
cd ironssis/trunk/test/ML_23Na_Al
../../../build/mulassis 23Na_-1.5_100-10000MeV_90.gmac
```

To execute the application from directory ironssis without a macrofile:

```
build/mulassis
```

Again, it is important to ensure all Geant4 environment variables are correctly defined in order to perform these runs (see http://geant4.cern.ch/support/download.shtml).

8 Validation and Verification Approach

8.1 Unit tests

Unit tests may be performed on the IRONSSIS software where it is appropriate to do so. Some visual inspection of the code shall also be performed of the code modules.

8.2 Compilation and Run tests in standalone operation

Compilation tests will be performed using appropriate compilers in a Linux environment, with the designated version(s) of Geant4, CLHEP and g++ (see section 10). Compiling and building the software will be performed using appropriate gmake and cmake scripts. The build process will be tested both for dynamically-linked libraries and statically-linked library operation.

Run tests of the software will be performed in a Linux environment. These will include inspection of the input and output files produced (see sections below).

NOTE: It is intended to build the IRONSSIS tool for the SEPEM environment, and therefore specific testing for the Windows operating system is not considered here.

8.3 Interface tests within the standalone environment

The standalone tests will be performed using a command-line input terminal.

Appropriate user input fields will be defined within different input files for IRONSSIS to provide comprehensive coverage of their possible combinations. Checks shall be made to verify that the appropriate internal variables are modified as a result of user input by one or a combination of:

- Print/write output of the variable, including echoing of the variable in the output files.
- Reviewing the variable during debugging of the executable.
- Checking that changing the variable has an appropriate impact on the response of the tool and the results generated.

Checks shall be made to confirm that the appropriate quantities are output to the report and CVS files generated by IRONSSIS and in the required format.

8.4 Interface tests within SEPEM environment

Appropriate user input fields will be defined through the SEPEM web-based GUI interface for IRONSSIS to provide comprehensive coverage of their possible combinations. Checks shall be made to verify that the appropriate internal variables are modified as a result of user input, by looking at the output files.

Checks shall be made to confirm that the appropriate quantities are output to the files generated by IRONSSIS and in the required format

8.5 Framework tests within SEPEM environment

Tests shall be performed to verify that the SEPEM interface software responds correctly to the user inputs and presents the appropriate web-based windows at each stage and expected outputs as provided by standalone version for same inputs.

8.6 Verification by comparisons with MathCAD results and validation of MathCAD algorithm with respect to MULASSIS Geant4 / FLUKA

Verification tests should be performed of the IRONSSIS against the MathCAD implementation of the algorithm for simple one-material shields. This should include the analysis of shielded ion fluences, TID, TNID and LET results for a few projectile and target conditions.

Validation tests should be performed of the IRONSSIS against the standalone version of MULASSIS to check that the simulation is performing as expected, and within the limits of the IRONSSIS algorithm implemented. This should include the analysis of shielded ion fluences, TID, TNID and LET results for a few projectile and target conditions.

8.7 Performance tests

If the software takes significant time to execute, performance profiling of the standalone version of IRONSSIS can be undertaken. This will help highlight where the computational process is spending most of the time, and whether this is excessive and the calculations could be accomplished in a more efficient manner. This could involve using the gprof tool with appropriate compilation options (-pg) for C code. Appropriate streamlining of the software will be identified and, if possible, implemented.

Note that in this instance performance profiling will focus on those aspects of new software developed as part of this project and the existing Geant4 toolkit will not be assessed due to the very significant size of undertaking such a task, and since the most appropriate individuals to judge the alternative methods of performing the different calculations within Geant4 are the toolkit authors.

The performance testing shall also include execution of the standalone software within a tool such as Valgrind (if this is practical) in order to check for memory leaks and rectify where needed.

9 Validation Results

9.1 Unit tests

No specific unit tests have been performed for IRONSSIS. Instead different aspects of IRONSSIS have been tested using different command input conditions to test the interface, correct response/calculation and output.

9.2 Compilation and build tests in standalone operation

Compilation and build in dynamic-link-library mode has been performed. There are no compile errors associated with the new code. Note that information about SVN revisions will be updated when software is formally submitted to the MULASSIS responsitory. Tests have yet to be performed for static library mode, but this is not expected to present a significant difficulty since MULASSIS can be compiled in this manner, and the IRONSSIS modifications do not introduce interfaces to new libraries.

```
ESHIEM/ironssis> ls
build  trunk
ESHIEM/ironssis> cd build
ironssis/build> cmake ../trunk
release_tag is [trunk]
SVN revision: 178
-- Configuring done
-- Generating done
-- Build files have been written to:
/home/geant4/ESAProjectWork/ESHIEM/ironssis/build
ironssis/build> make
[  2%] Building CXX object CMakeFiles/mulassis.dir/mulassis.cc.o
[  4%] Building CXX object CMakeFiles/mulassis.dir/src/SpenvisCSV.cc.o
[  6%] Building CXX object CMakeFiles/mulassis.dir/src/MLHisto1D.cc.o
[  8%] Building CXX object CMakeFiles/mulassis.dir/src/MLFluenceAnalyser.cc.o
[ 10%] Building CXX object CMakeFiles/mulassis.dir/src/IRRunMessenger.cc.o
[ 12%] Building CXX object CMakeFiles/mulassis.dir/src/MLMaterial.cc.o
[ 14%] Building CXX object CMakeFiles/mulassis.dir/src/IRFluenceAnalyser.cc.o
[ 16%] Building CXX object CMakeFiles/mulassis.dir/src/ParticlesBuilder.cc.o
[ 18%] Building CXX object CMakeFiles/mulassis.dir/src/MLSteppingAction.cc.o
[ 20%] Building CXX object CMakeFiles/mulassis.dir/src/IRNIELAnalyser.cc.o
[ 22%] Building CXX object CMakeFiles/mulassis.dir/src/IRRunManager.cc.o
[ 25%] Building CXX object CMakeFiles/mulassis.dir/src/MLSolarCellAnalyser.cc.o
/home/geant4/ESAProjectWork/ESHIEM/ironssis/trunk/src/MLSolarCellAnalyser.cc: In
function ‘CSVofstream& operator<<(CSVofstream&, MLSolarCellAnalyser&)’:
/home/geant4/ESAProjectWork/ESHIEM/ironssis/trunk/src/MLSolarCellAnalyser.cc:376:11
: warning: variable ‘n’ set but not used [-Wunused-but-set-variable]
/home/geant4/ESAProjectWork/ESHIEM/ironssis/trunk/src/MLSolarCellAnalyser.cc: In
function ‘RPTofstream& operator<<(RPTofstream&, MLSolarCellAnalyser&)’:
/home/geant4/ESAProjectWork/ESHIEM/ironssis/trunk/src/MLSolarCellAnalyser.cc:536:11
: warning: variable ‘n’ set but not used [-Wunused-but-set-variable]
[ 27%] Building CXX object CMakeFiles/mulassis.dir/src/MLRunManager.cc.o
[ 29%] Building CXX object CMakeFiles/mulassis.dir/src/MLEventAction.cc.o
[ 31%] Building CXX object CMakeFiles/mulassis.dir/src/MLGeometryMessenger.cc.o
[ 33%] Building CXX object CMakeFiles/mulassis.dir/src/MLAnalysisMessenger.cc.o
[ 35%] Building CXX object CMakeFiles/mulassis.dir/src/MLColour.cc.o
[ 37%] Building CXX object CMakeFiles/mulassis.dir/src/MLSD.cc.o
[ 39%] Building CXX object CMakeFiles/mulassis.dir/src/MLAnalysisManager.cc.o
[ 41%] Building CXX object CMakeFiles/mulassis.dir/src/VariableLengthPartition.cc.o
[ 43%] Building CXX object CMakeFiles/mulassis.dir/src/SplineInterpolation.cc.o
[ 45%] Building CXX object CMakeFiles/mulassis.dir/src/IRDoseAnalyser.cc.o
[ 47%] Building CXX object CMakeFiles/mulassis.dir/src/MLPHSAnalyser.cc.o
[ 50%] Building CXX object CMakeFiles/mulassis.dir/src/MLPhysicsUtil.cc.o
[ 52%] Building CXX object CMakeFiles/mulassis.dir/src/MLMaterialMessenger.cc.o
[ 54%] Building CXX object CMakeFiles/mulassis.dir/src/MLVersion.cc.o
[ 56%] Building CXX object CMakeFiles/mulassis.dir/src/IRAnalysisMessenger.cc.o
[ 58%] Building CXX object CMakeFiles/mulassis.dir/src/IRVersion.cc.o
[ 60%] Building CXX object CMakeFiles/mulassis.dir/src/MLDoseAnalyser.cc.o
[ 62%] Building CXX object CMakeFiles/mulassis.dir/src/MLEventActionMessenger.cc.o
[ 64%] Building CXX object CMakeFiles/mulassis.dir/src/MLRunAction.cc.o
[ 66%] Building CXX object CMakeFiles/mulassis.dir/src/IRLambdaInterpolation.cc.o
[ 68%] Building CXX object CMakeFiles/mulassis.dir/src/MLNIELAnalyser.cc.o
[ 70%] Building CXX object CMakeFiles/mulassis.dir/src/MLDoseEqAnalyser.cc.o
[ 72%] Building CXX object CMakeFiles/mulassis.dir/src/SpenvisCSVCollection.cc.o
[ 75%] Building CXX object
CMakeFiles/mulassis.dir/src/IRDEDXAndRangeInterpolation.cc.o
[ 77%] Building CXX object
CMakeFiles/mulassis.dir/src/IRDiffSpectrumInterpolation.cc.o
[ 79%] Building CXX object CMakeFiles/mulassis.dir/src/MLColourMessenger.cc.o
[ 81%] Building CXX object CMakeFiles/mulassis.dir/src/Histo1DVar.cc.o
[ 83%] Building CXX object CMakeFiles/mulassis.dir/src/MLGRASPhysicsList.cc.o
[ 85%] Building CXX object CMakeFiles/mulassis.dir/src/IRLayerFluxDose.cc.o
[ 87%] Building CXX object CMakeFiles/mulassis.dir/src/IRAnalysisManager.cc.o
[ 89%] Building CXX object CMakeFiles/mulassis.dir/src/IRPhysicsUtil.cc.o
[ 91%] Building CXX object CMakeFiles/mulassis.dir/src/MLGeometryConstruction.cc.o
[ 93%] Building CXX object
CMakeFiles/mulassis.dir/src/MLGRASPhysicsListMessenger.cc.o
[ 95%] Building CXX object CMakeFiles/mulassis.dir/src/MLHit.cc.o
[ 97%] Building CXX object CMakeFiles/mulassis.dir/src/MLNIELFunction.cc.o
[100%] Building CXX object
CMakeFiles/mulassis.dir/src/MLPrimaryGeneratorAction.cc.o
Linking CXX executable mulassis
[100%] Built target mulassis
ironssis/build>
```

9.3 Interface tests within the standalone environment

Interface tests have been performed running the example tests within the directory ironssis/trunk/test. In particular to test the response to new interface comments:

- All examples invoke /ironssis/beamOn to commence the fluence and dose calculation based on the parameters provided.
- ML_23Na_Al/23Na_-1.5_100-10000MeV_90_IFTest.gmac tests the definition of the integration parameters (commented /ironssis/integration comments in macro can be uncommented to see response to out-of-range inputs) and provides a list of the results. Also this example includes a change in the global error propagated into the results CSV and RPT files.
- ML_23Na_Al+NIEL and ML_p_Al+NIEL test the response to user input to define a NIEL function.
- For macrofile ML_23Na_Al/23Na_-1.5_100-10000MeV_90_IFTest.gmac, variously comment and uncomment the adjacent commands:
  - #/geometry/layer/shape sphere
  - /geometry/layer/shape slab

  - #/analysis/fluence/type PLANAR
  - /analysis/fluence/type OMNI

  - #/gps/particle e-

- /gps/particle ion
- /gps/ion 11 23 11 0.000000E+00

- #/gps/ang/type iso
- /gps/ang/type planar

These tests verify the response of IRONSSIS to geometry, fluence-tallying, primary particle species and angular conditions which are out of the scope of the calculation.

For these tests, the response of the software is correct and the parameters are updated or the execution of the run has been shown to stop appropriately and gracefully depending upon the input conditions.

9.4 Interface tests within SEPEM environment

(These tests will be undertaken and completed after IRONSSIS is integrated within SEPEM.)

9.5 Framework tests within SEPEM environment

(These tests will be undertaken and completed after IRONSSIS is integrated within SEPEM.)

9.6 Verification by comparisons with MathCAD results and validation of MathCAD algorithm with respect to MULASSIS Geant4 / FLUKA

Comparisons have been made between results from the non-MC IRONSSIS tool for treating simple shielding, and MULASSIS v1.25 (using Geant4 v9.6). In some cases, results are also presented from an implementation of the non-MC algorithm within MathCAD (MCAD), and Monte Carlo results from FLUKA. The examples shown here are a few of the comparisons for the case of:

- 1mm thick Al shields, or 1mm Al followed by 2mm C shields.
- Normal and cosine-law incidence.
- ^{23}Na ions and proton as incident projectiles

For the results presented here, a simple power-law spectra ($\psi(E)=10^{8}E^{-1.5}$ $cm^{-2}MeV^{-1}$) has been used. Some simulations were performed considering the attenuation of ions from nuclear interactions, although the impact of this on the observed fluxes is small for the shields considered. The figures below summarise a few of the results and show the shielded ^{23}Na or proton fluence after the Al or Al+C shields. For the power-law spectrum, the incident particles cover the energy range of 100 to 10000MeV – for ^{23}Na ions, the lower energy range will stop in 1mm Al, whilst 10000MeV ^{23}Na will propagate through 1mm Al + 2mm C.

The IRONSSIS results for the ^{23}Na appear to be in very good agreement with the MULASSIS and FLUKA simulations for the single Al shield (Figure 3 and Figure 4) and in reasonable agreement for the thicker shield (Figure 5 and Figure 6) although the low-energy statistics could benefit from further MC simulations (currently set to 10^{6}).

The IRONSSIS and MULASSIS proton fluence results for the power-law spectrum (Figure 7) do very nearly overlay each other, although this is perhaps a little misleading due to the energy of the source. Figure 8 shows the results for a flat and lower-energy proton source, and again the level of agreement is extremely good.

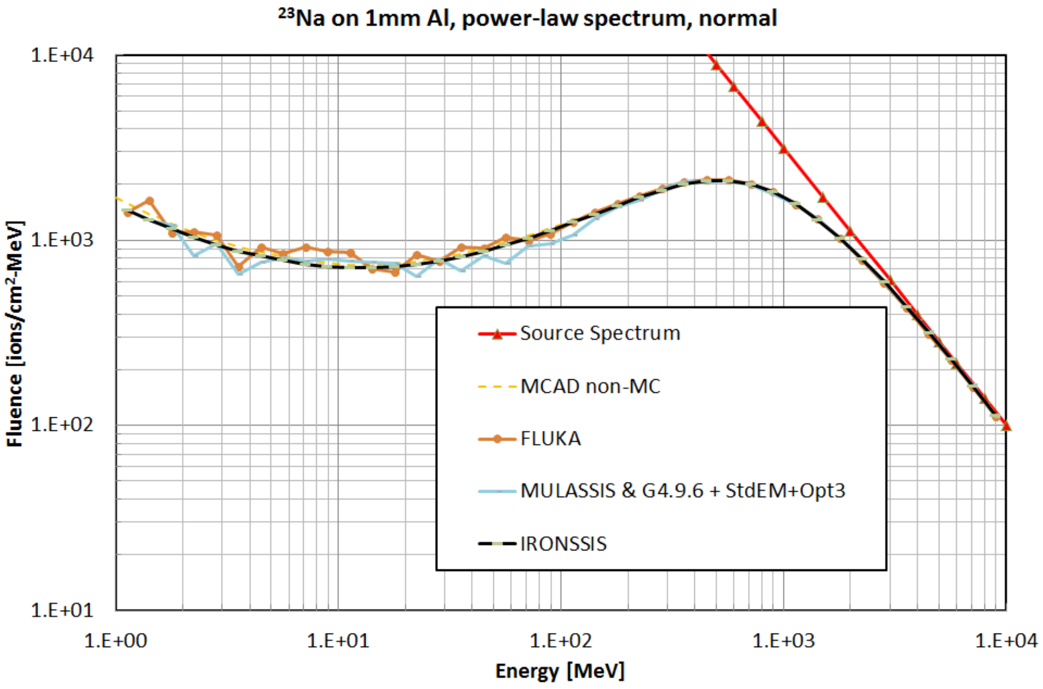


Figure 3: ^{23}Na incident upon 1mm Al normal incidence. Power-law spectrum.

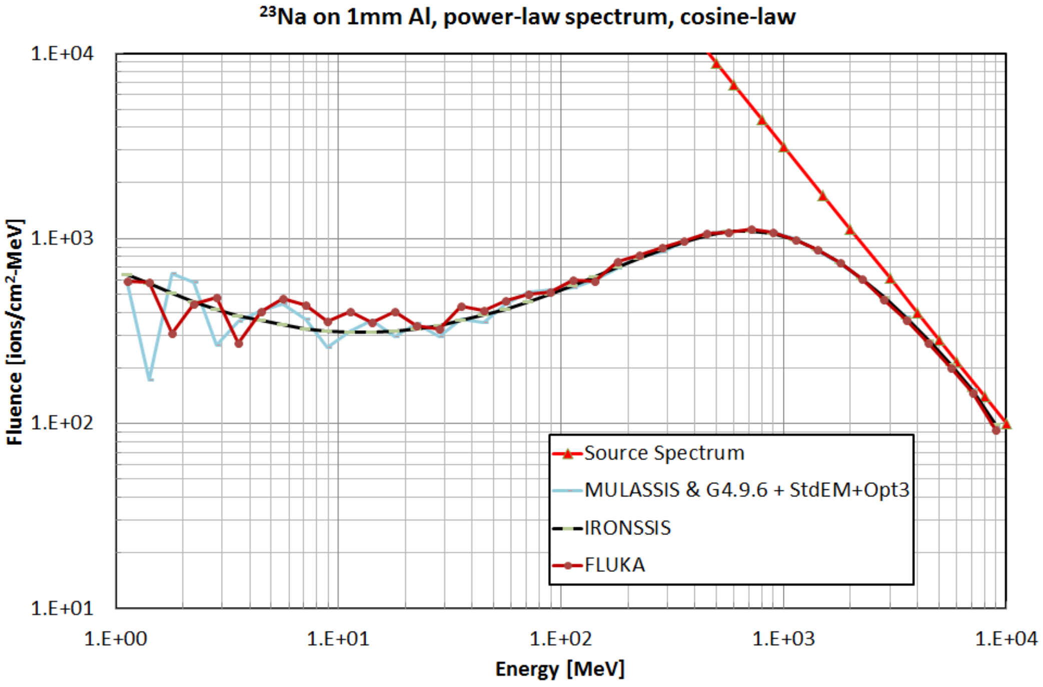


Figure 4: ^{23}Na incident upon 1mm Al cosine-law incidence. Power-law spectrum.

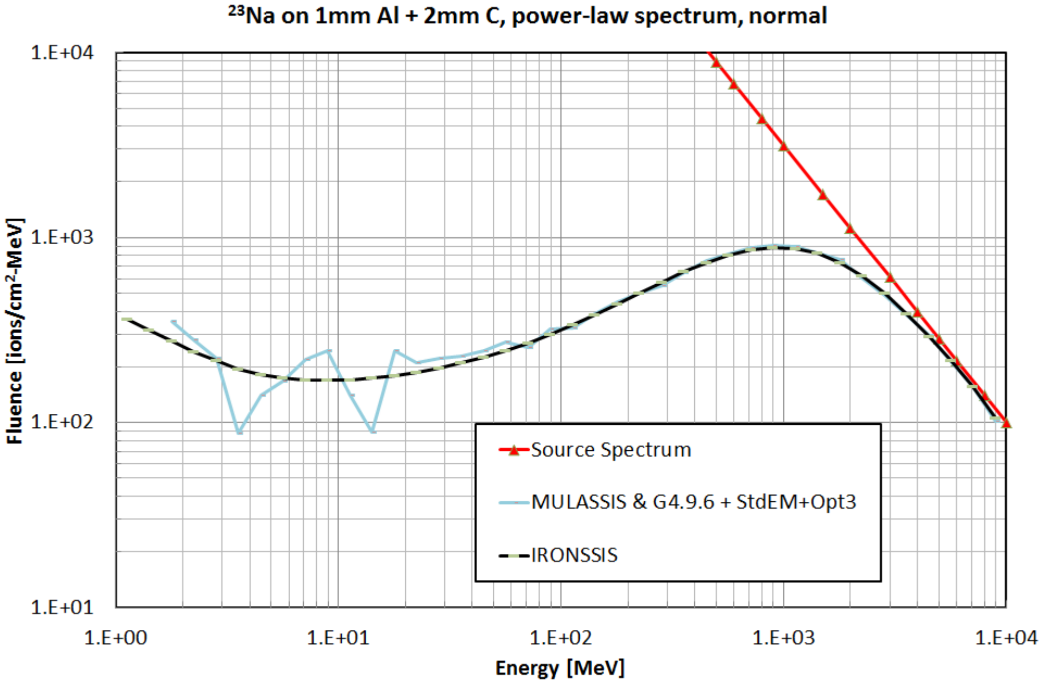


Figure 5: ^{23}Na incident upon 1mm Al + 2mm C normal incidence. Power-law spectrum. Graph shows shielded fluence after C layer.

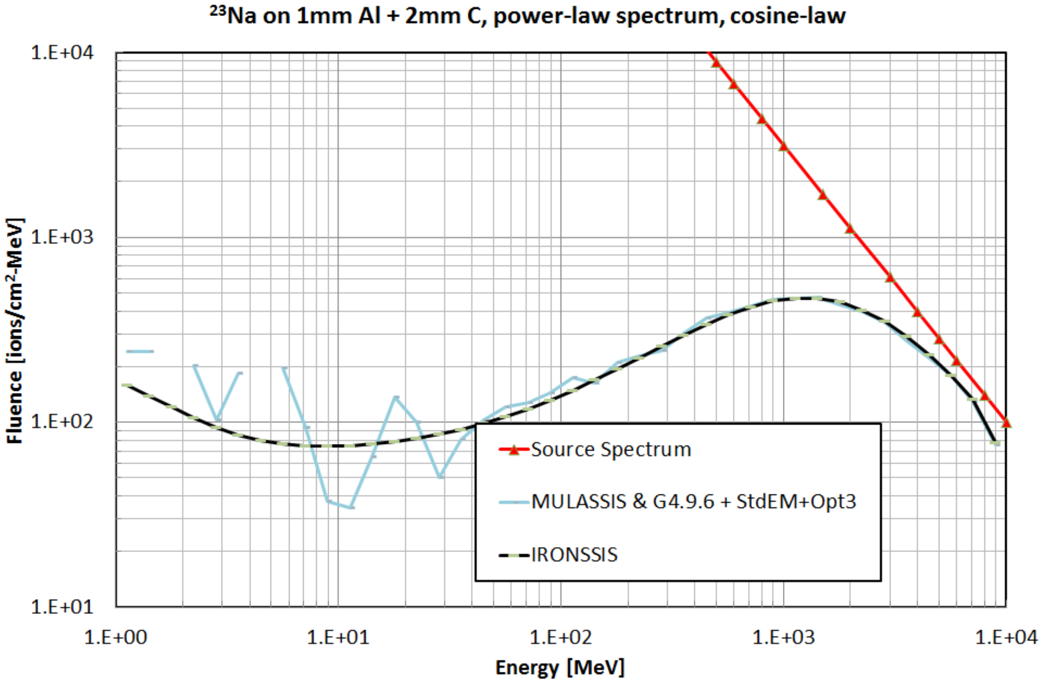


Figure 6: ^{23}Na incident upon 1mm Al + 2mm C cosine-law incidence. Power-law spectrum. Graph shows shielded fluence after C layer.

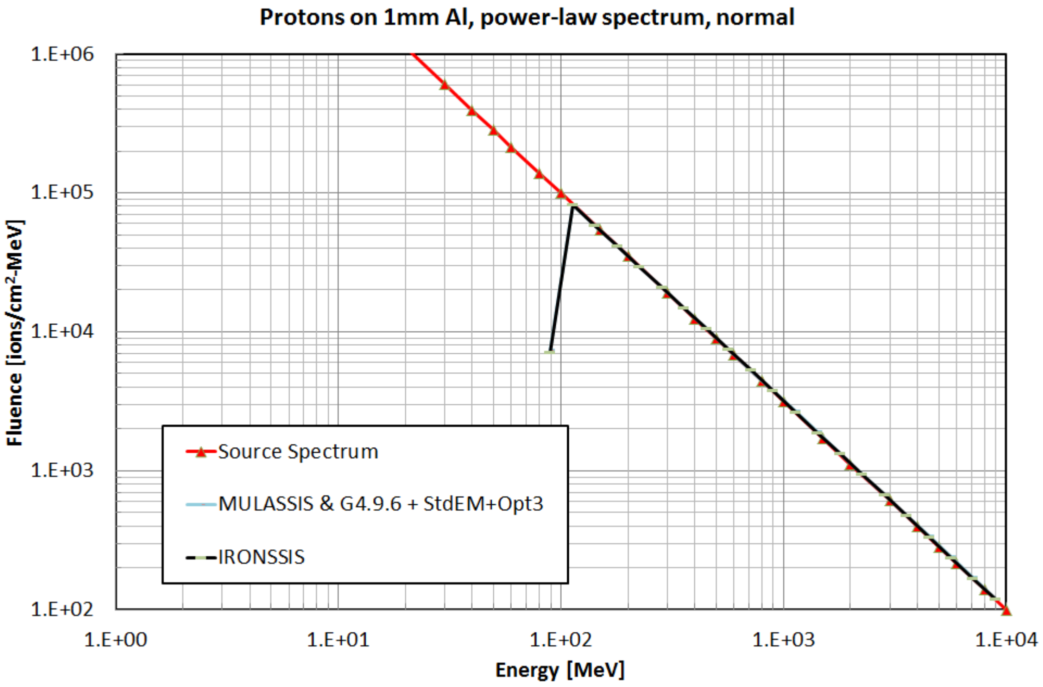


Figure 7: Protons upon 1mm Al normal incidence. Power-law spectrum.

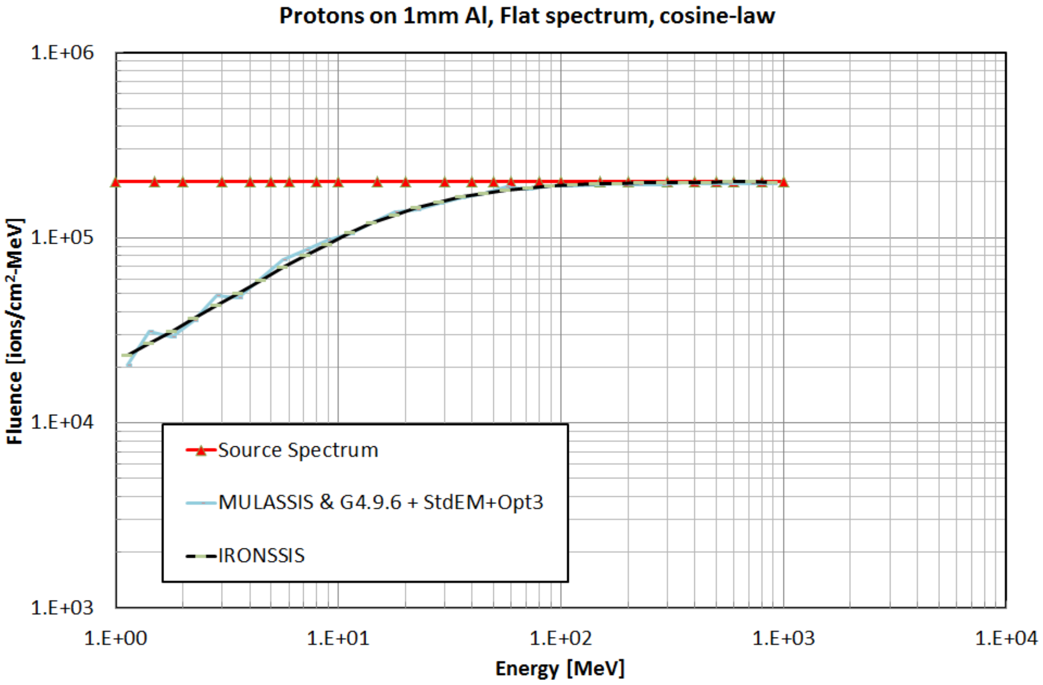


Figure 8: Protons upon 1mm Al cosine-law incidence. Square spectrum from 1 MeV to 1000 MeV.

Table 20 and Table 22 present the results from the TID and TNID analysis for these and some other irradiation scenarios (*e.g.* 45^o unidirectional, and cosine-law spanning 0^o to 45^o only). The results are taken directly from the CSV output files from IRONSSIS and MULASSIS without additional normalisation. Note that, strictly speaking, the TID data are energy-depositions in each layer per unit fluence, as the results have not been normalised by the areal mass of the shields.

Ion	Energy	Angle	Target	IRONSSIS Edep [MeV] (non-MC)	MULASSIS Edep [MeV] (MC)	Additional Information
23Na	Power-law 10^2 - 10^4	Normal	Al 1mm	270	271.7±0.2	ML_23Na_Al/23Na_-1.5_100-10000MeV_90: em
23Na	Power-law 10^2 - 10^4	45^o	Al 1mm	308	310.6 ± 0.2	ML_23Na_Al/23Na-1.5_100-10000MeV_45: em
23Na	Power-law 10^2 - 10^4	Cosine 0-90^o	Al 1mm	331	332 ± 0.3	ML_23Na_Al/23Na_-1.5o_100-10000MeV: em
23Na	Power-law 10^2 - 10^4	Cosine 0-45^o	Al 1mm	287	286.9 ± 0.2	(Modified ML_23Na_Al/23Na_-1.5o_100-10000MeV): em
23Na	Power-law 10^2 - 10^4	Normal	Al 1mm C 2mm	270 120	272.1 ± 0.3 121.5 ± 0.4	ML_23Na_Al+C/23Na_-1.5_100-10000MeV_90: em
23Na	Power-law 10^2 - 10^4	45^o	Al 1mm C 2mm	308 130	310.8 ± 0.4 132.7 ± 0.5	ML_23Na_Al+C/23Na_-1.5_100-10000MeV_45: em
23Na	Power-law 10^2 - 10^4	Cosine 0-90^o	Al 1mm C 2mm	331 135	331.5 ± 0.5 141.7 ± 0.6	ML_23Na_Al+C/23Na_-1.5_100-10000MeV_45: em
23Na	Power-law 10^2 - 10^4	Normal	Al 1mm	272	271.8 ± 0.2	ML_23Na_Al+nuc/23Na_-1.5_100-10000MeV_90: With nucl phys
23Na	Power-law 10^2 - 10^4	45^o	Al 1mm	311	310.7 ± 0.2	ML_23Na_Al+nuc/23Na_-1.5_100-10000MeV_45: With nucl phys
p+	Power-law 10^2 - 10^4	Normal	Al 1mm	0.699	0.7867 ± 0.0004	ML_p_Al/p_-1.5_100-10000MeV_90: em
p+	Power-law 10^2 - 10^4	45^o	Al 1mm	0.990	1.1162 ± 0.0005	ML_p_Al/p_-1.5_100-10000MeV_45: em
p+	Power-law 10^2 - 10^4	Cosine 0-90^o	Al 1mm	0.820	0.9211 ± 0.0004	ML_p_Al/p_-1.5o_100-10000MeV: em
p+	Flat, 1-1000	Cosine 0-90^o	Al 1mm	1.60	1.764 ± 0.003	ML_p_Al/p_sqo_1-1000MeV: em

Table 20: Comparison of "TID" results. Energy deposition from ionisation in layers (MeV per unit fluence)

Ion	Energy [MeV]	Angle	Target	IRONSSIS Dose [MeV/g] (non-MC)	MULASSIS Dose [MeV/g] (MC)	Additional Information
p+	Power-law 10^2 - 10^4	Normal	Al 1mm	1.73E-03	1.72E-03	ML_p_Al/p_-1.5_100-10000MeV_90:em, CERN NIEL
p+	Power-law 10^2 – 10^4	45°	Al 1mm	2.44E-03	2.44E-03	ML_p_Al/p_-1.5_100-10000MeV_45: em, CERN NIEL
p+	Power-law 10^2 - 10^4	Cosine 0-90°	Al 1mm	2.03E-03	2.02E-03	ML_p_Al/p_-1.5o_100-10000MeV: em, CERN NIEL
p+	Flat, 1-1000	Cosine 0-90°	Al 1mm	3.60E-03	4.02E-03	ML_p_Al/p_sqo_1-1000MeV: em, CERN NIEL
p+	Power-law 10^2 - 10^4	Normal	Al 1mm	4.32E-03	(N/A)	ML_p_Al+NIEL/p_-1.5_100-10000MeV_90:em, user-defined NIEL = 2.5 x CERN NIEL
p+	Power-law 10^2 – 10^4	45°	Al 1mm	6.12E-03	(N/A)	ML_p_Al/p_-1.5_100-10000MeV_45: em, user-defined NIEL = 2.5 x CERN NIEL
p+	Power-law 10^2 - 10^4	Cosine 0-90°	Al 1mm	5.06E-03	(N/A)	ML_p_Al/p_-1.5o_100-10000MeV: em, user-defined NIEL = 2.5 x CERN NIEL

Table 21: Comparison of TNID results. Non-ionising dose deposited after layers in Si (MeV/g per unit fluence).

The level of agreement for the “TID” and TNID data is seen to be extremely good for ^{23}Na, and so to the TNID data from the proton irradiations. For proton energy deposition results and TNID at lower energies are still shown to be very good agreement (better than 15% difference), despite the low mass of the protons and the relative simplicity of the non-MC calculations in IRONSSIS.

To test the performance of the interface to IRONSSIS that allows user definition a NIEL function, user-defined NIEL data were input to the calculation which represents 2.5 times the CERN NIEL function for protons (see Table 21, rows 6, 7 and 8 are for the user-defined NIEL compared with rows 2, 3 and 4 which use the CERN proton NIEL data). Since the IRONSSIS TNID results for these runs are 2.5x the standard results using the CERN proton NIEL, the interface and algorithm appears to operate as expected.

9.7 Performance tests

The speed of the IRONSSIS calculation is <5 seconds for most of the examples provided with the software (treating layers up to 3mm). One example treating 8 layers and a full user-defined NIEL function is completed within 12 seconds. SR 17 requires the results to be generated within 60 seconds. The equivalent runs within the conventional MULASSIS Monte Carlo simulation is several minutes to tens of minutes. Whilst there are potential improvements to be gained within the efficiency of the IRONSSIS non-MC numerical integration, these are not critical in defining the overall duration of the calculations within SEPEM, and therefore assessment of improvements will be undertaken later in the project.

Segmentation faults have not occurred either during execution of IRONSSIS or near termination of the IRONSSIS application. Valgrind tests have been performed and do not identify memory leaks arising from the new code during execution. However, there are memory leaks identified during termination of the program associated with the deletion (or non-deletion) of IRONSSIS, MULASSIS, or Geant4 objects. For the new code, conflicts in the deletion of MLNIELAnalyser and IRNIELAnalyser objects may cause memory leaks during termination of the program. Whilst this does not affect execution of the software and the results generated, and it has not been flagged as an error, some further analysis should be performed to determine a means by which this can be resolved satisfactorily.

10 Software Configuration Control

10.1 Configured items

Table 22 summaries the versions of the different software to be used in conjunction with the released version of IRONSSIS.

Category	Software package / files	Version
Operating system	Linux	3.1.0-1.2
Linux distribution	SuSE Linux	12.1
Compilers	gcc / g++	4.6.2
Other build S/W	cmake	tested with v2.8.6, but should be compatible with almost all versions
Original code	MULASSIS	1.25
Libraries	GEANT4	4.9.6 patch 02
	CLHEP	2.1.3.1
	ROOT	(N/A)

Table 22: Software configuration control list.

11 List of Abbreviations and Acronyms

ACE	Advanced Composition Explorer spacecraft
ASCII	American Standard Code for Information Interchange
CDF	cumulative (probability) distribution function
CSDA	continuous slowing-down approximation
CSV	comma-separated value (file format)
ESA	European Space Agency
ESHIEM	Energetic Solar Heavy Ion Environment Models Project
ESP	Energetic Solar Proton model
FPSAM	Fast Physical Shielding Analysis Model for SEPEM
GCR	galactic cosmic radiation galactic cosmic ray
Geant4	Geometry and Tracking code version 4
GPS	Geant4 General Particle Source
GRAS	Geant4 Radiation Analysis for Space
JPL	Jet Propulsion Laboratory – NASA Facility Model for solar energetic particle environment prediction
IRONSSIS	Ion Rapid One-dimensional Shielding Simulation Software
LEO	low-Earth orbit
LET	linear energy transfer
MC / non-MC	Monte Carlo / non-Monte-Carlo
MULASSIS	Multi-Layered Shielding Simulation Software
NASA	National Aeronautics and Space Administration
NIEL	non-ionising energy loss
NIST	National Institute of Standards and Technology
PDF	probability distribution function
PSAM	Physical Shielding Analysis Model for SEPEM (this definition includes MULASSIS)
RPP	rectangular parallelepiped
SEE	single event effect
SEP	solar energetic particles
SEPE	solar energetic particle event
SEU	single event upset
SPE	Solar particle event
SPENVIS	Space Environment Information System
SRIM	Stopping and Range of Ions in Matter code
SV	sensitive volume

TID	total ionising dose
TNID	total non-ionising dose

12 References

[1] "Energetic solar heavy-ion environment models," ESA Statement of Work, TEC-EES, T704-302EE, Issue 1, Rev 0, ITT AO 1-7131/12/NL/GLC, 4 April 2012.

[2] "Energetic Solar Heavy-Ion Environment Models (ESHIEM)," Kallisto Consultancy Proposal KALLISTO/BID/12003, June 2012.

[3] Pete Truscott, "ESHIEM Project: User Requirements Document, " Kallisto Consultancy document KALLISTO/URD/12020, February 2013.

[4]

[5] Fan Lei, "ESHIEM Project System Requirement Specification", V1.0, RadMod Research Ltd, RADMOD/SRD/201301, 2013.

[6] Fan Lei, "ESHIEM Project System Update Specifications," V1.0 RadMod Research Ltd, RADMOD/TN/201302, 2013.

[7] Cedric. Bonevie, "SEPEM Project – Design Definition File" , version R1.3, BISA, 01-February 2011.

[8] Norma Crosby *et al* "Solar Energetic Particle Environment Modeling (SEPEM) Project: Final report," version 1.B, September 2011.

[9] SEPEM web-site, http://dev.sepem.oma.be, 2013.

[10] Geant4 web-site: http://geant4.cern.ch, 2008.

[11] S Agostinelli *et al*, "Geant4 – a simulation toolkit," *Nucl Instrum Meth Phys Res*, **A506**, p250-303, 2003.

[12] MULASSIS website, 2012, [Online]. Available: http://reat.space.qinetiq.com/mulassis/

[13] F Lei, P R Truscott, C S Dyer, B Quaghebeur, D Heynderickx, P Nieminen, H Evans, and E Daly, "MULASSIS – a Geant4-based multilayered shielding simulation tool," IEEE Trans Nucl Sci, 49, No 6, pp2788-2793, 2002.

[14] Pete Truscott, "ESHIEM Project User Requirements Document," KALLISTO/URD/12020, June 2013.

[15] "Stopping of ions heavier than helium," International Commission on Radiation Units and Measurements (ICRU) Report 73, 2005.

[16] P Sigmund and A Schinner, "Binary theory of electronic stopping," Nucl Instrum Meth, B195, pp64-90, 2002.

[17] H Paul and A Schinner, "An empirical approach to the stopping power of solids and gasses for ions from 3Li to 18Ar", Nucl Instrum Meth, B179, pp299-315, 2001.

[18] Pete Truscott, "Rapid Reverse Monte Carlo Project: Review of Ion Physics Models, Requirements and Available Experimental Data," QinetiQ Technical Note QINETIQ/IS/AERO/TN0803256, November 2008

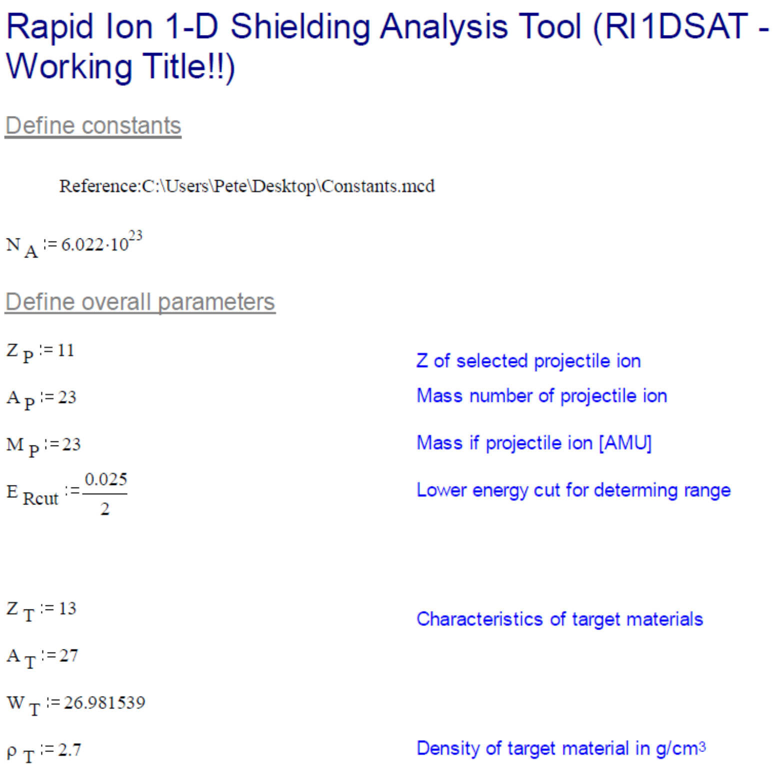
[19] R K Tripathi, Francis A Cucinotta, John W Wilson, NASA Technical Paper 3621, 1997.

[20] R K Tripathi, F A Cucinotta, and J W Wilson, "Universal parameterization of absorption cross-sections - Light systems," NASA Technical Paper TP-1999-209726, 1999.

[21]

A Appendix A: MathCAD Script for testing IRONSSIS

The following pages show the script which has been implemented into MathCAD and used for developing and testing the algorithm for a slab shield made of a single material. The specific parameters chosen are for ^{23}Na ions incident upon aluminium.



Rapid Ion 1-D Shielding Analysis Tool (RI1DSAT - Working Title!!)

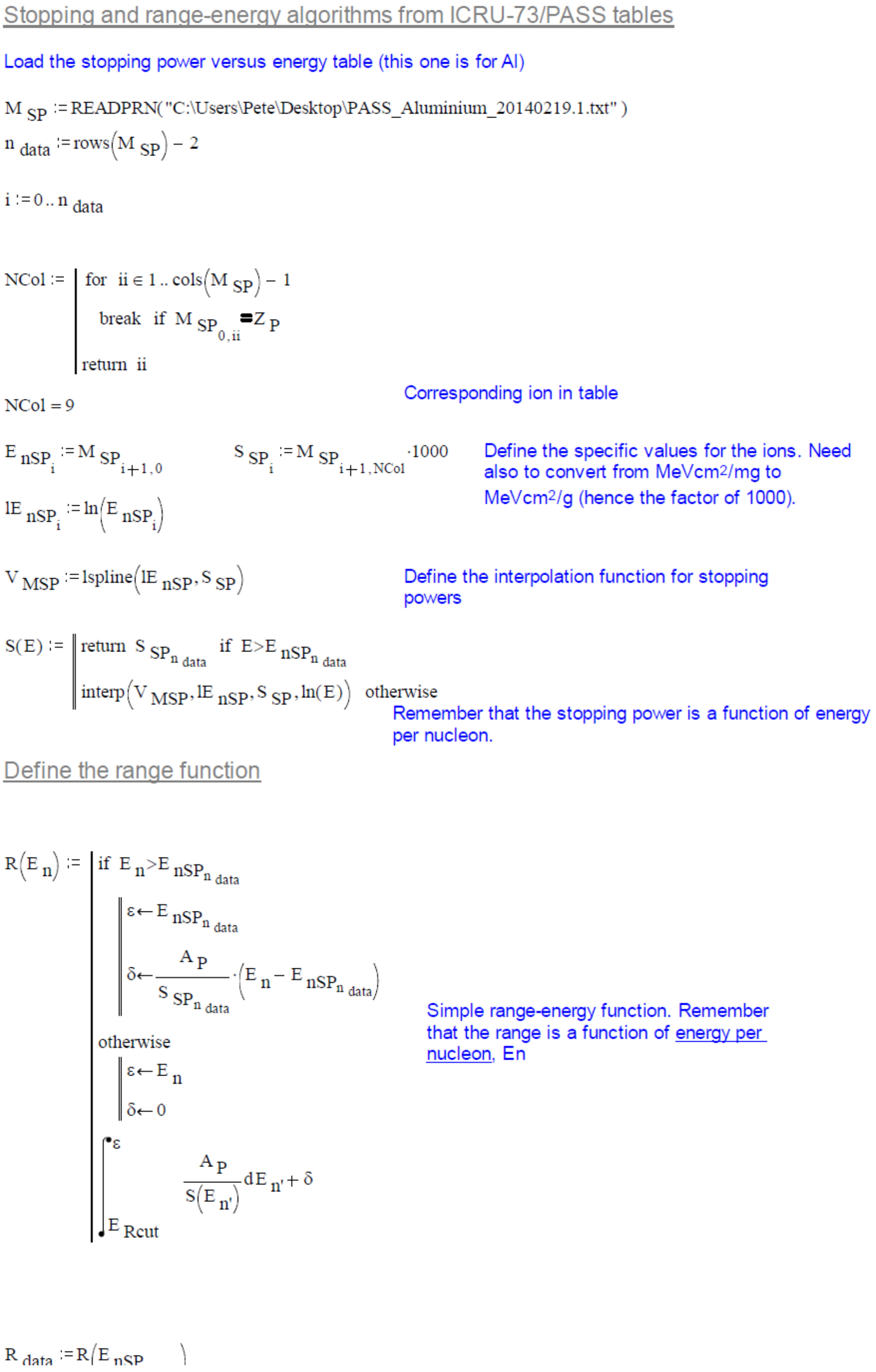
Define constants

Reference:C:\Users\Pete\Desktop\Constants.mcd

$N_A := 6.022 \cdot 10^{23}$

Define overall parameters

$Z_P := 11$	Z of selected projectile ion
$A_P := 23$	Mass number of projectile ion
$M_P := 23$	Mass if projectile ion [AMU]
$E_{Rcut} := \frac{0.025}{2}$	Lower energy cut for determing range
$Z_T := 13$	Characteristics of target materials
$A_T := 27$	
$W_T := 26.981539$	
$\rho_T := 2.7$	Density of target material in g/cm3



Stopping and range-energy algorithms from ICRU-73/PASS tables

Load the stopping power versus energy table (this one is for Al)

$M_{SP} :=$ READPRN("C:\Users\Pete\Desktop\PASS_Aluminium_20140219.1.txt")

$n_{data} := \text{rows}(M_{SP}) - 2$

$i := 0..n_{data}$

```
$NCol :=$ | for  $ii \in 1..\text{cols}(M_{SP}) - 1$
          |     break  if  $M_{SP_{0,ii}} = Z_P$
          | return  ii
```

Corresponding ion in table

$NCol = 9$

$E_{nSP_i} := M_{SP_{i+1,0}}$ $\qquad S_{SP_i} := M_{SP_{i+1,NCol}} \cdot 1000$

Define the specific values for the ions. Need also to convert from MeVcm^2/mg to MeVcm^2/g (hence the factor of 1000).

$lE_{nSP_i} := \ln(E_{nSP_i})$

$V_{MSP} := \text{lspline}(lE_{nSP}, S_{SP})$

Define the interpolation function for stopping powers

```
$S(E) :=$ | return  $S_{SP_{n_{data}}}$  if  $E > E_{nSP_{n_{data}}}$
          | $\text{interp}(V_{MSP}, lE_{nSP}, S_{SP}, \ln(E))$  otherwise
```

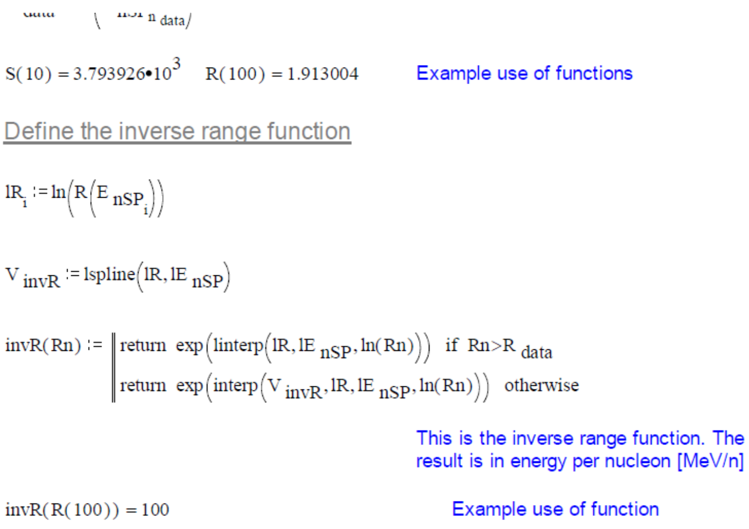
Remember that the stopping power is a function of energy per nucleon.

Define the range function

```
$R(E_n) :=$ | if  $E_n > E_{nSP_{n_{data}}}$
            |     | $\varepsilon \leftarrow E_{nSP_{n_{data}}}$
            |     | $\delta \leftarrow \frac{A_P}{S_{SP_{n_{data}}}} \cdot (E_n - E_{nSP_{n_{data}}})$
            | otherwise
            |     | $\varepsilon \leftarrow E_n$
            |     | $\delta \leftarrow 0$
            | $\int_{E_{Rcut}}^{\varepsilon} \frac{A_P}{S(E_{n'})} dE_{n'} + \delta$
```

Simple range-energy function. Remember that the range is a function of energy per nucleon, En

$R_{data} := R(E_{nSP}\quad)$



$S(10) = 3.793926 \cdot 10^{3} \quad R(100) = 1.913004$ Example use of functions

Define the inverse range function

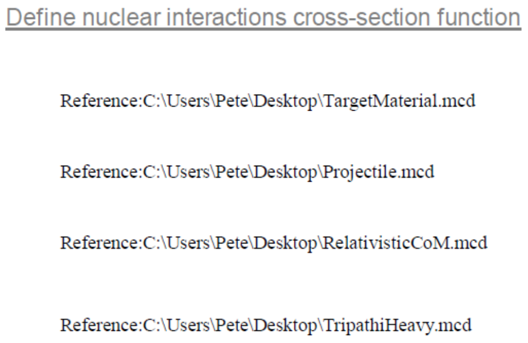
$lR_i := \ln\left(R\left(E_{nSP_i}\right)\right)$

$V_{invR} := \text{lspline}\left(lR, lE_{nSP}\right)$

$$invR(Rn) := \left|\begin{array}{l} \text{return } \exp\left(\text{linterp}\left(lR, lE_{nSP}, \ln(Rn)\right)\right) \text{ if } Rn > R_{data} \\ \text{return } \exp\left(\text{interp}\left(V_{invR}, lR, lE_{nSP}, \ln(Rn)\right)\right) \text{ otherwise} \end{array}\right.$$

This is the inverse range function. The result is in energy per nucleon [MeV/n]

$invR(R(100)) = 100$ Example use of function



Define nuclear interactions cross-section function

Reference:C:\Users\Pete\Desktop\TargetMaterial.mcd

Reference:C:\Users\Pete\Desktop\Projectile.mcd

Reference:C:\Users\Pete\Desktop\RelativisticCoM.mcd

Reference:C:\Users\Pete\Desktop\TripathiHeavy.mcd

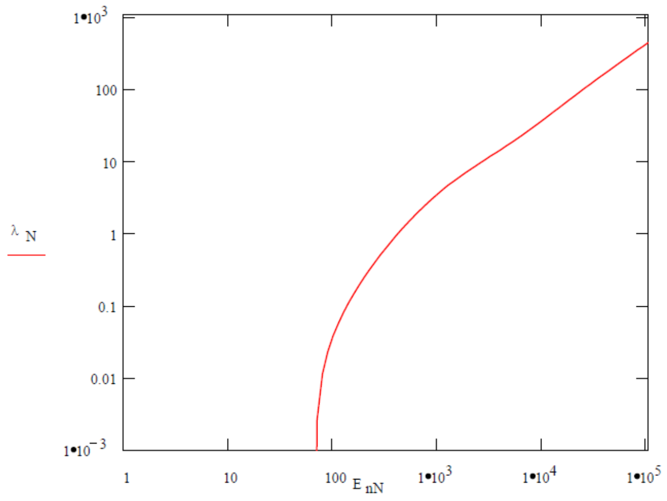
Definition of nuclear attenuation function

$i2 := 0..100 \qquad E_{nNlo} := 1 \qquad E_{nNhi} := 10^{5}$

$$E_{nN_{i2}} := E_{nNlo} \cdot 10^{\frac{i2}{20}}$$

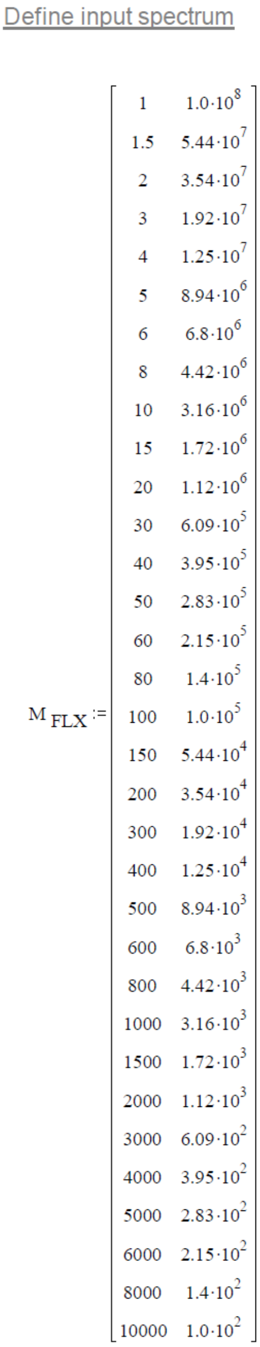
$$P(E_n) := \frac{\sigma_{TH}(E_n)}{S(E_n)}$$

$$\lambda_{N_{i2}} := \left| \begin{array}{l} \lambda \leftarrow \dfrac{N_A \cdot A_P}{W_T} \cdot 10^{-27} \cdot \left[\displaystyle\int_{E_{nNlo}}^{E_{nN_{i2}}} P(E'_n)\, dE'_n \right] \\ \text{return } \lambda \text{ if } \lambda \geq 10^{-10} \\ \text{return } 10^{-10} \text{ otherwise} \end{array} \right.$$



$lE_{nN_{i2}} := \ln\left(E_{nN_{i2}}\right) \qquad l\lambda_{N_{i2}} := \ln\left(\lambda_{N_{i2}}\right)$

$\lambda(E_n) := \exp\left(\text{linterp}\left(lE_{nN}, l\lambda_{N}, \ln(E_n)\right)\right) \qquad \lambda\left(\dfrac{100000}{A_P}\right) = 15.119038$



Define input spectrum

$$M_{FLX} := \begin{bmatrix} 1 & 1.0\cdot 10^{8} \\ 1.5 & 5.44\cdot 10^{7} \\ 2 & 3.54\cdot 10^{7} \\ 3 & 1.92\cdot 10^{7} \\ 4 & 1.25\cdot 10^{7} \\ 5 & 8.94\cdot 10^{6} \\ 6 & 6.8\cdot 10^{6} \\ 8 & 4.42\cdot 10^{6} \\ 10 & 3.16\cdot 10^{6} \\ 15 & 1.72\cdot 10^{6} \\ 20 & 1.12\cdot 10^{6} \\ 30 & 6.09\cdot 10^{5} \\ 40 & 3.95\cdot 10^{5} \\ 50 & 2.83\cdot 10^{5} \\ 60 & 2.15\cdot 10^{5} \\ 80 & 1.4\cdot 10^{5} \\ 100 & 1.0\cdot 10^{5} \\ 150 & 5.44\cdot 10^{4} \\ 200 & 3.54\cdot 10^{4} \\ 300 & 1.92\cdot 10^{4} \\ 400 & 1.25\cdot 10^{4} \\ 500 & 8.94\cdot 10^{3} \\ 600 & 6.8\cdot 10^{3} \\ 800 & 4.42\cdot 10^{3} \\ 1000 & 3.16\cdot 10^{3} \\ 1500 & 1.72\cdot 10^{3} \\ 2000 & 1.12\cdot 10^{3} \\ 3000 & 6.09\cdot 10^{2} \\ 4000 & 3.95\cdot 10^{2} \\ 5000 & 2.83\cdot 10^{2} \\ 6000 & 2.15\cdot 10^{2} \\ 8000 & 1.4\cdot 10^{2} \\ 10000 & 1.0\cdot 10^{2} \end{bmatrix}$$

$j := 0 .. \mathrm{rows}\left(M_{FLX}\right) - 1$

$E_{FLX_j} := M_{FLX_{j,0}}$ $\phi_{FLX_j} := M_{FLX_{j,1}}$

$lE_{FLX_j} := \ln\left(E_{FLX_j}\right)$ $l\phi_{FLX_j} := \ln\left(\phi_{FLX_j}\right)$

$V_{MFLX} := \mathrm{lspline}\left(lE_{FLX}, l\phi_{FLX}\right)$ $\phi(E) := \exp\left(\mathrm{interp}\left(V_{MFLX}, lE_{FLX}, l\phi_{FLX}, \ln(E)\right)\right)$

This expression above is no longer used

$\phi(E) := \exp\left(\mathrm{linterp}\left(lE_{FLX}, l\phi_{FLX}, \ln(E)\right)\right)$

$E_{ilo} := E_{FLX_0}$ $E_{ihi} := 100000$

Define output spectrum energies

$k := 0..90 \qquad E_{\psi lo} := 1$

$E_{o_k} := E_{\psi lo} \cdot 10^{\frac{k}{20}}$

Information about the material

$T := 0.27$ Thickness of material in g/cm2

$E_{nlim} := \mathrm{invR}(T) \qquad E_{nlim} = 32.36728$

General expression for spectrum as a function of energy and angle

$$\psi(E,\theta) := \left| \begin{array}{l} T_{eff} \leftarrow \dfrac{T}{\cos(\theta)} \\ E_{ni} \leftarrow \mathrm{invR}\left(R\left(\dfrac{E}{A_P}\right) + T_{eff}\right) \\ \mathrm{return}\ \ 0\ \ \mathrm{if}\ E_{ni} \cdot A_P > E_{ihi} \\ \kappa \leftarrow \dfrac{S(E_{ni})}{S\left(\dfrac{E}{A_P}\right)} \\ \mathrm{return}\ \ \phi(E_{ni} \cdot A_P) \cdot \kappa \ \ \mathrm{otherwise} \end{array} \right.$$

$$\psi_{nuc}(E,\theta) := \left| \begin{array}{l} T_{eff} \leftarrow \dfrac{T}{\cos(\theta)} \\ E_{ni} \leftarrow \mathrm{invR}\left(R\left(\dfrac{E}{A_P}\right) + T_{eff}\right) \\ \mathrm{return}\ \ 0\ \ \mathrm{if}\ E_{ni} \cdot A_P > E_{ihi} \\ \kappa \leftarrow \dfrac{S(E_{ni})}{S\left(\dfrac{E}{A_P}\right)} \\ \xi \leftarrow \exp\left(\lambda\left(\mathrm{invR}\left(R\left(\dfrac{E}{A_P}\right)\right)\right) - \lambda(E_{ni})\right) \\ \mathrm{return}\ \ \xi \cdot \phi(E_{ni} \cdot A_P) \cdot \kappa \ \ \mathrm{otherwise} \end{array} \right.$$

The following is just for some checks of the calculation process

$EE := E_{o_{90}} \qquad EE = 3.162278 \cdot 10^4 \qquad \dfrac{EE}{A_P} = 1.374903 \cdot 10^3$

$EE_{ni} := \mathrm{invR}\left(R\left(\dfrac{EE}{A_P}\right) + T\right) \qquad EE_{ni} = 1.351329 \cdot 10^3$

$\lambda\left(\dfrac{EE}{A_P}\right) = 4.937196 \qquad \lambda(EE_{ni}) = 4.844423 \qquad R\left(\dfrac{EE}{A_P}\right) = 116.707414$

$\exp\left(\lambda\left(\dfrac{EE}{A_P}\right) - \lambda(EE_{ni})\right) = 1.097212 \qquad \mathrm{invR}\left(R\left(\dfrac{EE}{A_P}\right)\right) = 1.349083 \cdot 10^3$

Output spectrum for normally incident particles

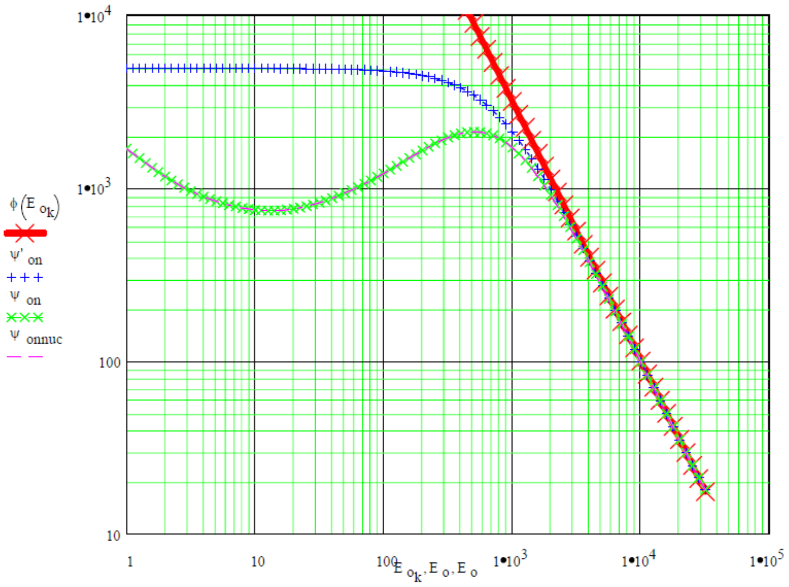
$\psi_{on_k} := \psi\left(E_{o_k}, 0\right)$

$\psi_{onnuc_k} := \psi_{nuc}\left(E_{o_k}, 0\right)$

$$\kappa_k := \frac{S\left(invR\left(R\left(\frac{E_{o_k}}{A_P}\right)+T\right)\right)}{S\left(\frac{E_{o_k}}{A_P}\right)}$$

ψ_{on}

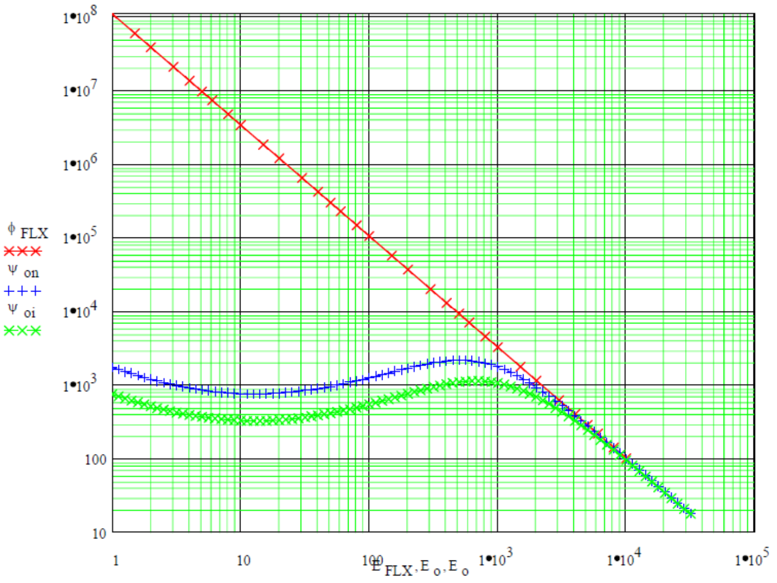
$$\psi'_{on_k} := \frac{\psi_{on_k}}{\kappa_k}$$



Output spectrum for isotropically incident particles

$$\Psi_{oi_k} := \int_0^{\frac{\pi}{2}\cdot 0.999} \Psi\left(E_{o_k},\theta\right)\cdot\sin(\theta)\,d\theta$$

Check if this is normalised correctly!!!!!



$\Psi_{oi} =$

Initial distribution list

Name	Organisation	Copy type/#
Dr Pete Truscott	Kallisto Consultancy	SC, HC #3 & #4
Dr Daniel Heynderickx	DH Consultancy	SC
Dr Fan Lei	RadMod Research	SC
Dr Athina Varotsou	TRAD	SC
Dr Piers Jiggens	ESA/ESTEC TEC-EES	SC, HC #1 & #2
Dr Alain Hilgers	ESA/ESTEC TEC-EES	SC
Hugh Evans	ESA/ESTEC TEC-EES	SC

(HC indicates hardcopy and copy number, SC indicates softcopy)