

SIMNRA User's Guide

Matej Mayer

Max-Planck-Institut für Plasmaphysik
Boltzmannstr. 2
85748 Garching
Germany

matej.mayer@ipp.mpg.de
Tel.: +49 89 32 99 16 39
Fax.: +49 89 32 99 22 79
www.simnra.com
mam.home.ipp.mpg.de

This document describes SIMNRA version 7.04

© Max-Planck-Institut für Plasmaphysik, 2024

Abstract

This document describes the use of the program SIMNRA and the physical concepts implemented therein. SIMNRA is a Microsoft Windows program for the simulation of charged particle energy spectra and gamma ray yields for ion beam analysis with MeV ions. SIMNRA simulates charged particle energy spectra with Rutherford scattering cross-sections (RBS), non-Rutherford cross-sections (EBS), nuclear reactions (NRA), elastic recoil detection analysis (ERDA) with light and heavy incident ions, and medium energy ion scattering (MEIS). It simulates also particle induced gamma ray emission (PIGE) yields.

SIMNRA can calculate spectra for any ion-target combination including incident heavy ions and any geometry including backscattering, forward scattering, and transmission geometries. Arbitrary multi-layered foils in front of the detector and arbitrary multi-layered windows (such as exit windows for an external beam) in front of the sample can be used.

Several different stopping power data sets are available. Energy loss straggling is calculated including the corrections by Chu and Yang to Bohr's theory, energy loss straggling propagation in thick layers is taken into account. Additionally the effects of multiple small-angle and plural large angle scattering can be calculated approximately as well as surface and substrate roughness. Many non-Rutherford and nuclear reaction cross-sections for incident protons, deuterons, ^3He -, ^4He -, ^6Li - and ^7Li -ions are included. Data fitting (layer thicknesses, compositions etc.) is possible by means of the Simplex algorithm.

OLE automation allows automated analysis of large numbers of spectra and the development of new applications. SIMNRA supports the IBA data format (IDF) for data storage and exchange of data between different simulation programs.

SIMNRA is easy to use due to the graphical Microsoft Windows user interface: SIMNRA makes full use of the graphics capacities of Windows. SIMNRA shell extensions extend the abilities of Windows Explorer and Windows Search, thus allowing a seamless integration of SIMNRA into the Windows system.

Publications

Additional publications about SIMNRA. The first is the original technical report describing SIMNRA and should be used as general reference for the program in all publications. All other references describe various aspects of the program.

- M. Mayer. *SIMNRA User's Guide*. Tech. rep. IPP 9/113. Garching: Max-Planck-Institut für Plasmaphysik, 1997. URL: <https://mam.home.ipp.mpg.de/Report%20IPP%209-113.pdf>
- M. Mayer. "SIMNRA, a Simulation Program for the Analysis of NRA, RBS and ERDA". in: *Proceedings of the 15th International Conference on the Application of Accelerators in Research and Industry*. Ed. by J. L. Duggan and I.L. Morgan. Vol. 475. AIP Conference Proceedings. Woodbury, New York: American Institute of Physics, 1999, p. 541. DOI: [10.1063/1.59188](https://doi.org/10.1063/1.59188). URL: [https://mam.home.ipp.mpg.de/Mayer-AIP-Conference-Proceedings-475-\(1999\)-541.pdf](https://mam.home.ipp.mpg.de/Mayer-AIP-Conference-Proceedings-475-(1999)-541.pdf)
- W. Eckstein and M. Mayer. "Rutherford backscattering from layered structures beyond the single scattering model". In: *Nucl. Instr. Meth. B* 153 (1999), p. 337. DOI: [10.1016/S0168-583X\(98\)01011-8](https://doi.org/10.1016/S0168-583X(98)01011-8)
- M. Mayer. "Ion Beam Analysis of Rough Thin Films". In: *Nucl. Instr. Meth. B* 194 (2002), p. 177. DOI: [10.1016/S0168-583X\(02\)00689-4](https://doi.org/10.1016/S0168-583X(02)00689-4). URL: [https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-194-\(2002\)-177.pdf](https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-194-(2002)-177.pdf)
- M. Mayer, K. Arstila, K. Nordlund, E. Edelmann, and J. Keinonen. "Multiple scattering of MeV ions: Comparison between the analytical theory and Monte-Carlo and molecular dynamics simulations". In: *Nucl. Instr. Meth. B* 249 (2006), p. 823. DOI: [10.1016/j.nimb.2006.03.148](https://doi.org/10.1016/j.nimb.2006.03.148). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0027-0B39-C>. URL: [https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-249-\(2006\)-823.pdf](https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-249-(2006)-823.pdf)
- N.P. Barradas, K. Arstila, G. Battistig, M. Bianconi, N. Dytlewski, et al. "International Atomic Energy Agency intercomparison of ion beam analysis software". In: *Nucl. Instr. Meth. B* 262 (2007), p. 281. DOI: [10.1016/j.nimb.2007.05.018](https://doi.org/10.1016/j.nimb.2007.05.018). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0027-0732-B>. URL: [https://mam.home.ipp.mpg.de/Barradas-Nuclear-Instruments-Methods-B-262-\(2007\)-281.pdf](https://mam.home.ipp.mpg.de/Barradas-Nuclear-Instruments-Methods-B-262-(2007)-281.pdf)
- M. Mayer, K. Arstila, and U. von Toussaint. "Skewness of energy-loss straggling and multiple-scattering energy distributions". In: *Nucl. Instr. Meth. B* 268 (2010), p. 1744. DOI: [10.1016/j.nimb.2010.02.057](https://doi.org/10.1016/j.nimb.2010.02.057). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0026-F07D-B>. URL: [https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-268-\(2010\)-1744.pdf](https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-268-(2010)-1744.pdf)

- N.P. Barradas, M. Mayer, and M. Thompson. “A new ion beam analysis data format”. In: *Nucl. Instr. Meth. B* 268 (2010), p. 1824. DOI: [10.1016/j.nimb.2010.02.093](https://doi.org/10.1016/j.nimb.2010.02.093). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0026-EED0-D>. URL: [https://mam.home.ipp.mpg.de/Barradas-Nuclear-Instruments-Methods-B-268-\(2010\)-1824.pdf](https://mam.home.ipp.mpg.de/Barradas-Nuclear-Instruments-Methods-B-268-(2010)-1824.pdf)
- H. Langhuth, M. Mayer, and S. Lindig. “Layer Morphology Analysis of Sputter-eroded Silicon Gratings using Rutherford Backscattering”. In: *Nucl. Instr. Meth. B* 269 (2011), p. 1811. DOI: [10.1016/j.nimb.2011.05.002](https://doi.org/10.1016/j.nimb.2011.05.002). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0026-EAC7-7>. URL: [https://mam.home.ipp.mpg.de/Langhuth-Nuclear-Instruments-Methods-B-269-\(2011\)-1811.pdf](https://mam.home.ipp.mpg.de/Langhuth-Nuclear-Instruments-Methods-B-269-(2011)-1811.pdf)
- M. Mayer, W. Eckstein, H. Langhuth, F. Schiettekatte, and U. von Toussaint. “Computer Simulation of Ion Beam Analysis: Possibilities and Limitations”. In: *Nucl. Instr. Meth. B* 269 (2011), p. 3006. DOI: [10.1016/j.nimb.2011.04.066](https://doi.org/10.1016/j.nimb.2011.04.066). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0026-E993-3>. URL: [https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-269-\(2011\)-3006.pdf](https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-269-(2011)-3006.pdf)
- M. Mayer, U. von Toussaint, J. Dewalque, O. Dubreuil, C. Henrist, et al. “Rutherford backscattering analysis of porous thin TiO₂ films”. In: *Nucl. Instr. Meth. B* 273 (2012), p. 83. DOI: [10.1016/j.nimb.2011.07.045](https://doi.org/10.1016/j.nimb.2011.07.045). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0026-E4F3-2>. URL: [https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-273-\(2012\)-83.pdf](https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-273-(2012)-83.pdf)
- M. Mayer. “Improved Physics in SIMNRA 7”. In: *Nucl. Instr. Meth. B* 332 (2014), p. 176. DOI: [10.1016/j.nimb.2014.02.056](https://doi.org/10.1016/j.nimb.2014.02.056). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0023-C6B3-9>. URL: [https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-332-\(2014\)-176.pdf](https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-332-(2014)-176.pdf)
- N. Pessoa Barradas, J. Cruz, M. Fonseca, A.P. de Jesus, A. Lagoyannis, et al. “International Atomic Energy Agency inter-comparison of particle induced gamma-ray emission codes for bulk samples”. In: *Nucl. Instr. Meth. B* 468 (2020), p. 37. DOI: [10.1016/j.nimb.2020.02.019](https://doi.org/10.1016/j.nimb.2020.02.019)
- M. Mayer. “An expert-assisted system for improving the quality of IBA simulations by SIMNRA”. in: *J. Phys.: Conf. Ser.* 2326 (2022), p. 012007. DOI: [10.1088/1742-6596/2326/1/012007](https://doi.org/10.1088/1742-6596/2326/1/012007). URL: [https://mam.home.ipp.mpg.de/Mayer-Journal-Physics-Conference-Series-2326-\(2022\)-012007.pdf](https://mam.home.ipp.mpg.de/Mayer-Journal-Physics-Conference-Series-2326-(2022)-012007.pdf)

DOI: Link to the original version of the publication at the journal pages. You may need institutional access in order to view the full text.

eDoc: Link to the full text of the publication at the eDoc server of the Max Planck Society. The text is available within the [open access initiative of the Max Planck Society](#). The text is identical to the original publication, but has been reformatted due to copyright reasons.

URL: Link to the full text of the publication at the SIMNRA home page. The text is freely available. The text is identical to the eDoc version.

Contents

Publications	iii
Contents	v
Registration and payment information	xi
Product license agreement	xii
Version history	xiii
1. Overview	1
1.1. Organization of this document	1
1.2. Conventions in this manual	1
2. Installation	2
2.1. System requirements	2
2.2. Installation	2
2.3. Uninstalling SIMNRA	7
3. Using SIMNRA	8
3.1. Basic steps	8
3.2. General usage notes	9
3.2.1. Opening files	9
3.2.2. Data input forms	9
3.2.3. Toolbar	9
3.2.4. Statusbar	10
3.2.5. Internationalization	10
3.2.6. High resolution devices	10
3.3. File menu	11
3.3.1. New	11
3.3.2. Open	11
3.3.3. Save	11
3.3.4. Save as	11
3.3.5. Read Spectrum Data	12
3.3.6. Write Spectrum Data	17
3.3.7. RUMP	17
3.3.8. Print	18
3.3.9. Exit	18
3.4. Edit menu	19
3.4.1. Copy data	19
3.4.2. Copy page	19

Contents

3.5.	Setup menu	20
3.5.1.	Experiment	20
3.5.2.	Experiment: More Options	24
3.5.3.	Calculation	30
3.6.	Target menu	39
3.6.1.	Target	39
3.6.2.	Foil	44
3.6.3.	Window	44
3.7.	Reactions menu	46
3.7.1.	Origin of cross-section data files	47
3.7.2.	Replacement of cross section data files	47
3.8.	Calculate menu	48
3.8.1.	Calculate Spectrum	48
3.8.2.	Calculate Spectrum Fast	48
3.8.3.	Fit Spectrum	48
3.8.4.	Kinematics	51
3.8.5.	Stopping	51
3.8.6.	Cross Section	51
3.8.7.	Density	52
3.8.8.	Particles*sr	52
3.8.9.	Exit Angle Beta	52
3.8.10.	Subtract Pile-up	52
3.9.	Tools menu	54
3.9.1.	Data Reader	54
3.9.2.	Integrate Spectrum	54
3.9.3.	Nearest Elements	54
3.9.4.	Smoothed Experimental Spectrum	55
3.9.5.	chi2 evaluation	55
3.9.6.	Resize Spectrum	56
3.9.7.	Check Input	56
3.10.	Plot menu	57
3.11.	Options menu	59
3.11.1.	Create List of Cross-Sections	59
3.11.2.	Preferences	59
3.12.	Help menu	60
3.12.1.	User's Guide	60
3.12.2.	Home Page	60
3.12.3.	IBANDL Cross-Sections	60
3.12.4.	Mott Cross-Sections	60
3.12.5.	About...	60
3.12.6.	Register...	61
3.13.	PIGE	61
3.14.	Neutron-induced reactions and neutron-ERD	62

3.15. Data exchange with other programs	63
3.15.1. Graphics programs: Excel, Origin,	63
3.15.2. RUMP	63
3.15.3. IBA data furnace	65
3.16. Importing spectrum data in any format	66
3.17. Adding new cross-section data	67
3.17.1. The R33 file format	67
3.18. Using SRIM stopping powers	71
3.18.1. Trouble shooting	71
3.19. Using user-defined stopping powers	73
3.20. Energy calibration issues	74
3.20.1. Detector nonlinearity	74
3.20.2. Energy calibration for different ion species	74
3.21. Data smoothing	75
3.22. Programming support	76
3.22.1. Command line parameters	76
3.22.2. OLE automation	76
3.23. The xnra file format	76
4. Physics	79
4.1. Atomic data	81
4.2. Scattering kinematics	82
4.2.1. Elastic scattering	82
4.2.2. Nuclear reactions	82
4.3. Cross-section data	84
4.3.1. Rutherford cross-section	84
4.3.2. Screened Rutherford cross-sections	84
4.3.3. Non-Rutherford cross-sections	87
4.4. Number of backscattered particles	90
4.5. Evaluation of energy loss	93
4.6. Stopping power data	96
4.6.1. Andersen-Ziegler stopping	96
4.6.2. Ziegler-Biersack stopping	97
4.6.3. KKK stopping	99
4.6.4. SRIM stopping	99
4.6.5. Nuclear stopping	99
4.6.6. Stopping in compounds	100
4.7. Detector energy resolution	101
4.7.1. Time-of-flight detector	101
4.7.2. Electrostatic detector	101
4.8. Straggling	102
4.8.1. Overview	102
4.8.2. Electronic energy loss straggling	103
4.8.3. Nuclear energy loss straggling	111

Contents

4.8.4.	Energy loss straggling in compounds	113
4.8.5.	Geometrical straggling	113
4.9.	Multiple and plural scattering	116
4.9.1.	Overview	116
4.9.2.	Multiple (small angle) scattering	116
4.9.3.	Plural (large angle) scattering	118
4.10.	Skewness of energy distributions	120
4.10.1.	The two-piece normal distribution (TPND)	120
4.10.2.	Asymmetry parameter of arbitrary distributions	122
4.10.3.	Electronic energy-loss straggling	122
4.10.4.	Multiple scattering	123
4.11.	Surface roughness	130
4.11.1.	Rough film on a smooth substrate	130
4.11.2.	Smooth film on a rough substrate	134
4.11.3.	Arbitrary lateral structures	145
4.12.	Layer porosity	148
4.12.1.	The path length distribution function and its moments	148
4.12.2.	Relation between path length distribution and energy distribution	151
4.12.3.	Combining in- and outgoing paths	153
4.13.	Live time and pile-up	155
4.13.1.	Live time correction	155
4.13.2.	Calculation of pile-up	155
4.14.	Expert system	165
4.14.1.	Performed tests	166
5.	Examples	170
5.1.	RBS: Rutherford cross-sections	170
5.2.	RBS: Non-Rutherford cross-sections	173
5.3.	ERDA: Non-Rutherford cross-sections	175
6.	Acknowledgements	176
A.	OLE automation reference	178
A.1.	Startup mode	178
A.2.	Objects	178
A.3.	Properties and methods	179
A.3.1.	Simnra.App	179
A.3.2.	Simnra.Setup	181
A.3.3.	Simnra.Projectile	182
A.3.4.	Simnra.Calc	183
A.3.5.	Simnra.Target	184
A.3.6.	Simnra.Foil	186
A.3.7.	Simnra.Window	187
A.3.8.	Simnra.Fit	188

A.3.9. Simnra.Spectrum	188
A.3.10. Simnra.PIGE	189
A.3.11. Simnra.Stopping	189
A.3.12. Simnra.CrossSec	190
A.4. Data types	191
A.5. Simnra.App	191
A.5.1. Properties	191
A.5.2. Methods	203
A.6. Simnra.Setup	217
A.6.1. Properties	217
A.6.2. Methods	227
A.7. Simnra.Projectile	232
A.7.1. Properties	232
A.7.2. Methods	233
A.8. Simnra.Calc	234
A.8.1. Properties	234
A.8.2. Methods	245
A.9. Simnra.Target	248
A.9.1. Properties	248
A.9.2. Methods	258
A.10. Simnra.Foil	268
A.10.1. Properties	268
A.10.2. Methods	273
A.11. Simnra.Window	279
A.11.1. Properties	279
A.11.2. Methods	285
A.12. Simnra.Fit	290
A.12.1. Properties	290
A.12.2. Methods	295
A.13. Simnra.Spectrum	296
A.13.1. Input parameter	296
A.13.2. Properties	296
A.13.3. Methods	300
A.14. Simnra.PIGE	308
A.14.1. Properties	308
A.14.2. Methods	309
A.15. Simnra.Stopping	310
A.15.1. Input parameter	310
A.15.2. Methods	310
A.16. Simnra.CrossSec	314
A.16.1. Properties	314
A.16.2. Methods	318
A.17. Error handling	326

Contents

A.18. Programming examples	327
A.18.1. Using the OLE automation objects	327
A.18.2. Using the Fit object	328
A.18.3. Adding layers and elements	328
A.18.4. Using the Spectrum.GetDataArray object	332
B. SIMNRA windows shell extensions	333
B.1. Thumbnail Provider	333
B.2. Property handler	336
C. SIMNRA access keys and keyboard shortcuts	339
D. The R33 cross section file format	341
D.1. Introduction	341
D.2. The new R33 Format definition.	342
D.2.1. Syntax of an R33 Entry	343
D.2.2. List of legal entries	343
E. Origin fitting function for the two-piece normal distribution	348
E.1. Installation in Origin	348
E.2. Usage in Origin	348
Bibliography	349
Index	359

Registration and payment information

SIMNRA is not free. It is a shareware program: You can use SIMNRA for a trial period of thirty (30) days without fee. If you want to use SIMNRA after this period, you have to pay the registration fee and you have to register the program. See [SIMNRA > Registration and Pricing](#) for pricing details.

Each license is valid for one computer. Installation of SIMNRA on several computers requires several licenses.

To register the program and get a registration number either send an email, a letter by surface mail, or a fax to Dr. Matej Mayer. The address can be found on the title page of this manual. You will receive your registration number and an invoice with full payment details within a few days. Run the program and click *Help:Register...* to enter the registration number.

Most experimental cross-section data files are retrieved from the Ion Beam Analysis Nuclear Data Library (IBANDL) which is maintained and provided freely by the Nuclear Data Services of the IAEA (<http://www-nds.iaea.org>). IBANDL is a product of an IAEA Research Coordination Project and while all efforts were made to ensure that the most accurate information was adopted, the IAEA accepts no responsibility for the usage of IBANDL data. These files are not included in the shareware fee, but are freely available for scientific purposes.

The stopping power data files SC0EF .95A and SC0EF .95B have been taken from the SRIM 97 distribution.

SIMNRA was developed at the Max-Planck-Institut für Plasmaphysik, Garching, Germany.

Product license agreement

The product license agreement can be found in the file [LICENSE.TXT](#). The product license agreement is part of SIMNRA.

Version history

Recent program changes are documented in the file [Changes.txt](#).

1. Overview


1.1. Organization of this document

This document is organized in the following way:

- System requirements, installation and uninstallation of the program are described in [chapter 2](#).
- The use of the program is described in [chapter 3](#). A quick overview about the steps necessary to calculate a spectrum is given in [section 3.1](#). More details are found in the rest of [chapter 3](#).
- The physical concepts implemented in the program are described in detail in [chapter 4](#).
- Some examples for the abilities of the program are shown in [chapter 5](#).

1.2. Conventions in this manual

Links to sections, figures, pages, references, internet web sites and additional text files are highlighted in [blue](#). A click with the left mouse will bring you to the link destination.

Additional attachments are marked with the symbol . Attachments can be extracted to disc by clicking with the right mouse button and selecting *Save to disc...*

2. Installation

System requirements, the installation process of the program, directory and file structure, and the uninstallation process are described in this chapter. Registering the program is described on [page xi](#).

2.1. System requirements

- SIMNRA requires Windows 7 or higher.
- A screen resolution of 1024×768 pixels or higher is recommended. High dpi devices are supported, SIMNRA has been tested with 4 k displays up to 288 dpi (300% text size).
- SIMNRA requires about 160 MB free hard disk space.
- User's Guide and help system are in PDF format and require Adobe Acrobat or Adobe Reader/Adobe Acrobat Reader DC. Adobe Reader/Adobe Acrobat Reader DC can be obtained freely from the Adobe web site at www.adobe.com. SIMNRA has been tested with Adobe Reader X, Adobe Reader XI, Adobe Acrobat Reader DC and Adobe Acrobat Reader DC versions 2017 to 2021. The SIMNRA help system may have limited functionality with alternative PDF-viewers (such as Sumatra PDF).
- Reading spectrum data in Canberra's CAM-file format requires the Genie-2000 software package. See [section 3.3](#) for details.
- The use of SRIM stopping powers requires the software package SRIM 2003 or higher. SRIM can be downloaded freely from the SRIM home page, see [section 2.2](#) and [section 3.18](#) for details. It is recommended to install SRIM *before* installing SIMNRA, see [section 2.2](#).

2.2. Installation

SIMNRA is distributed with a setup program containing a 32- and a 64-bit version. The version is selected automatically: On 32-bit Windows systems the 32-bit version of SIMNRA is installed, and the 64-bit version on 64-bit Windows.

- The installation of SIMNRA requires administrator privileges.
- To install SIMNRA run the setup program and follow the instructions.

- A previous version of SIMNRA can be uninstalled automatically during the installation process. User-defined cross-section data files from the CrSec\User directory and user-defined stopping power data from the Stop directory can be moved automatically to the new installation during the installation process.
- The SIMNRA setup program will try to determine if SRIM is already installed on the computer. If SRIM is successfully detected, then SIMNRA is automatically configured for using SRIM stopping powers. This automatic detection requires installation of SRIM *before* installation of SIMNRA and works with SRIM 2008, but not with SRIM 2013¹. See [section 3.18](#) if you install SRIM *after* SIMNRA or if you use SRIM 2013.

After running the setup program you should have obtained the files listed in [Table 2.1](#).

To register run SIMNRA, click [Help: Register...](#) and enter your registration number in the appropriate field. Registering SIMNRA for *All users* of the computer requires running SIMNRA with administrator privileges by right-clicking the program and then selecting *Run as administrator*, see [subsection 3.12.6](#).

¹Unfortunately SRIM 2013 ignores all Microsoft recommendations for software installation.

Table 2.1.: Directory structure and files used by SIMNRA.

Directory	Files	
	Simnra.exe libgcc_s_sjlj-1.dll, libiconv-2.dll, libwinpthread-1.dll, libxml2- 2.dll, zlib1.dll Simnra.tlb Readme.txt Changes.txt Changes 7.0.txt Changes 6.0.txt Changes 5.0.txt Changes 4.4.txt License.txt SIMNRA-Users-Guide.pdf destination.txt	Executable program dynamic link libraries for XML support SIMNRA type library Readme file Describes the changes since version 7.0 Describes the changes from version 6.0 to 7.0 Describes the changes from version 5.0 to 6.0 Describes the changes from version 4.4 to 5.0 Describes the changes from version 3.0 to 4.4 Product license agreement This manual For internal use by the help system
	CanberraReadObject.exe ReadObject.exe SIMNRAThumbnailProvider.dll SIMNRAPropertyHandler.dll Simnra.propdesc prop.exe unins000.exe, unins000.dat	COM object for reading Canberra's CAM file format, see section 3.3.5 COM object for fast reading of nra and xnra file properties Thumbnail provider shell extension Property handler shell extension XML description of SIMNRA properties Program for registering/unregistering of SIMNRA properties Uninstaller for uninstalling SIMNRA
\Atom	AtomData.dat CoulombBarriers.dat	Atomic data Coulomb barriers from [122, Appendix 8]
<i>continued on next page</i>		

<i>continued from previous page</i>		
Directory	Files	
	MS_Tau_DPhi.dat	Precomputed values for multiple scattering calculations
\CrSec	Replace640.lst Replace.lst	Replacement of cross-section data files, see subsection 3.7.2 Replacement of cross-section data files, see subsection 3.7.2
\CrSec\IBANDL \CrSec\Additional \CrSec\User	*.r33 *.r33	Cross-section data files from IBANDL Additional cross-section data files Directory for user-supplied cross-section data files
\Default	Setup.xnra Calc.xnra	default experimental setup default parameters for calculations
\DLL	*.dll	dynamic link libraries used by SIMNRA
\DPASS	DPASS.exe, DPASS.pdf, DPASS_DB.dat	DPASS program for stopping power calculations
\Layers	*.lay	predefined materials (mylar, stainless steel)
\Logos	*.png	Logos for Windows 10 tiles
\MottCalc	*.*	MottCalc program for calculation of Mott cross sections
\Samples	*.nra	examples
\Stop	STOPH.DAT STOPHE.DAT LCORRHI.DAT SCOE.95A SCOE.95B CHU_CORR.DAT StopMax_H.dat, Stop- Max_He.dat SRIM2003_x_y_z.dat ZB_x_y_z.dat	Electronic stopping power data (Andersen/Ziegler) Electronic stopping power data (Ziegler/Biersack) Chu correction data to Bohr straggling Energies of stopping maxima for H and He ions Stopping power data files with SRIM 2003 stopping ^a Stopping power data files with Ziegler-Biersack stopping ^b
<i>continued on next page</i>		

^aThese files are only created whenever a calculation with SRIM 2003 stopping is performed. The number of these files depends on the number of already performed calculations.

^bThese files are only created whenever a calculation with Ziegler-Biersack stopping is performed. The number of these files depends on the number of already performed calculations.

<i>continued from previous page</i>		
Directory	Files	
\UserDLL	Sample*.dpr	Code examples in Pascal for user supplied dynamic link libraries, see section 3.16
\XML	IDF_Template.xml	XML template file for xnra files

2.3. Uninstalling SIMNRA

SIMNRA is shipped with an automatic uninstall program. Refer to your Microsoft Windows documentation on how to uninstall programs. SIMNRA should be *always* uninstalled using the uninstaller, as otherwise registry entries will be not uninstalled correctly.

3. Using SIMNRA

This chapter describes the use of SIMNRA. A quick overview about the basic steps necessary to calculate a backscattering spectrum can be found in [section 3.1](#). The use of data input forms and of the SIMNRA toolbar are explained in [subsection 3.2.2](#) and [subsection 3.2.3](#), language-specific questions (such as the decimal separator and file naming) are discussed in [subsection 3.2.5](#). The rest of [chapter 3](#) describes all menu items in detail. The underlying physics is described in [chapter 4](#).

3.1. Basic steps

This section gives a quick overview about the basic steps necessary to calculate a backscattering spectrum.

Three steps must be performed before a backscattering spectrum can be calculated: In a first step the experimental situation (incident ions, geometry) has to be defined, then the target must be created, and in a third step the cross-sections used for the calculation have to be chosen.

1. Click [Setup: Experiment](#). Choose the incident ions, the ions energy, define the scattering geometry (see [Figure 3.6](#)), and enter the energy calibration of the experiment.
2. Click [Target: Target](#). Here you create the target. Each target consists of *layers*. Each layer consists of different elements with some atomic concentration, which does not change throughout the layer, and each layer has a thickness.
3. If there is a foil in front of the detector, then click [Target: Foil](#) for the definition of a foil. The default is no foil in front of the detector. Like the target, the foil can consist of different layers, and the layers can have different compositions.
4. Click [Reactions](#). Choose which cross-section data are used for the simulation. The default are Rutherford cross-sections for all elements. You can select non-Rutherford cross-sections instead and you can add nuclear reactions.
5. Now the spectrum can be calculated. Click [Calculate: Calculate Spectrum](#) for a simulation of the spectrum.
6. With [Setup: Calculation](#) the parameters for the calculation can be altered. The default values are normally sufficient, and these values should be only changed if you know what you are doing.
7. With [File: Read Spectrum Data](#) a measured spectrum can be imported for comparison with the simulated one and for data fitting.

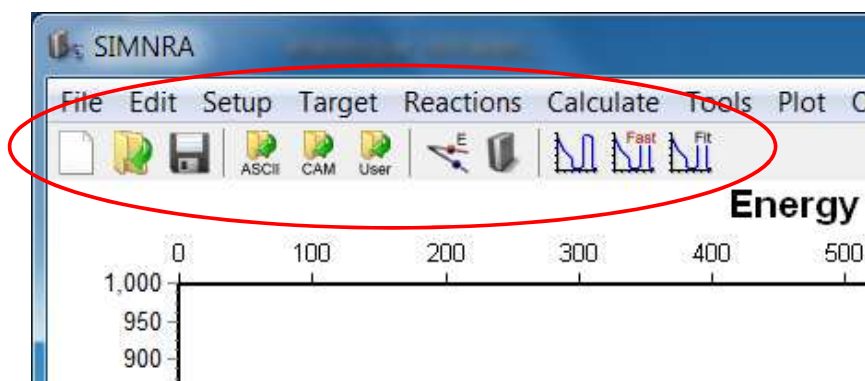


Figure 3.1.: SIMNRA toolbar, marked in red.

3.2. General usage notes

3.2.1. Opening files

SIMNRA data files in *nra* and *xnra* file format can be opened in different ways:

- By running SIMNRA and clicking *File: Open...*, see [page 11](#).
- By drag-and-drop of a *nra* or *xnra* file on a running instance of SIMNRA.
- By double-clicking a *nra* or *xnra* file in Windows Explorer.

3.2.2. Data input forms

Input data are entered in different forms. For example, clicking *Setup: Experiment* shows a window where experimental parameters are entered. These forms can remain open: Parameters can be modified and new calculations can be performed (by clicking *Calculate: Calculate spectrum* or the button in the [toolbar](#)) without the need to close the window, thus minimizing the number of necessary mouse clicks.

- Do not change parameters while a simulation is being calculated. This may result in unpredictable behavior and may crash the program!

Entering floating-point numbers depends on the regional settings of the computer, see [subsection 3.2.5](#).

3.2.3. Toolbar

Often used commands are accessible through a toolbar, see [Figure 3.1](#). Clicking a button is identical to navigating to the corresponding command in the menu. The toolbar can be switched off, see [section 3.11](#).

3. Using SIMNRA

3.2.4. Statusbar

The statusbar at the bottom of the program window displays information about the current status of the program, the progress of calculations, and displays the quadratic deviation between an experimental and a simulated spectrum¹. The χ^2 value is the total quadratic deviation in the regions of interest, the reduced χ_r^2 is χ^2 per channel, i.e. χ^2 divided by the number of channels in the regions of interest. A double-click on the χ^2 value allows to define the regions of interest where χ^2 is determined, see [subsection 3.9.5](#).

3.2.5. Internationalization

SIMNRA 6.51 and higher uses the regional settings of the installed Windows system. Depending on these settings either a decimal point '.' or a decimal comma ',' have to be used as decimal separator for floating point numbers in all data input forms. The Windows regional settings can be changed by opening *My Computer*, then *Control Panel*, and then *Regional Settings*.

For reading and writing files the regional settings are ignored, and the decimal separator is always a decimal point '.' for floating point numbers.

SIMNRA 6.70 and higher uses Unicode encoding for all text strings. File- and directory names can contain any Arabic, Cyrillic, Chinese, Thai etc. characters.

3.2.6. High resolution devices

SIMNRA is DPI-aware and runs correctly on high-resolution devices up to 4 k. It has been tested at desktop scaling factors from 100% (96 DPI) to 300% (288 DPI).

¹The χ^2 value is only displayed if an experimental spectrum and a simulated spectrum are present.

3.3. File menu

In the *File* menu all necessary commands for reading and saving files and data, printing spectra and terminating the program are located.

3.3.1. New

This menu item resets the program to its starting values. All calculated spectra, target, foil and setup definitions are deleted.

3.3.2. Open...

This menu item reads a saved calculation from disk. SIMNRA can open files in the following formats:

- *xnra file format*: SIMNRA 6.50 and higher use the new *xnra* file format. This file format is an xml-based format according to the IBA data format (IDF) definition [17]. See [section 3.23](#) for details of the *xnra* file format.
- *nra file format*: SIMNRA versions prior to 6.50 used the meanwhile outdated *nra* file format. Files in this format still can be opened, but will be converted to *xnra* files when saved.
- *IDF file format*: These are xml-files according to the IBA data format (IDF) definition [17]. This file format is very versatile for storing ion beam analysis (IBA) data and spectra and can be used for exchanging data between different IBA simulation programs.

3.3.3. Save

This menu item saves all current parameters, target and foil definitions, experimental and simulated data to disk. The data are saved as *xnra file*. The default file extension is *xnra*. See [section 3.23](#) for details of the *xnra* file format.

Depending on the settings in *Options: Preferences: Saving* (see [section 3.11](#)) the old file can be saved to a file named *backup.xnra*. In the case of erroneous overwriting of a file the old data can be recovered from this file.

3.3.4. Save as...

Like *Save*, but you will be prompted for the name of the file, and you can select the file format. The file can be saved in the following formats:

- *xnra file format*: The default file format is *xnra*, which is used by SIMNRA 6.50 and higher. This file format is an xml-based format according to the IBA data format (IDF) definition [17]. The default file extension is *xnra*. See [section 3.23](#) for details of the *xnra* file format.

3. Using SIMNRA

- *IDF file format*: Saves the file as xml-file according to the IBA data format (IDF) definition [17]. This file format is very versatile for storing ion beam analysis (IBA) data and spectra and can be used for exchanging data between different IBA simulation programs. The default file extension for IDF files is xml.

xnra-files are always valid IDF-files. Therefore by selecting either *xnra* or IDF, a file with the same content is written. Only the file extension is changed.

3.3.5. Read Spectrum Data

This menu item allows the import of experimental data. The availability of menu items depends on the settings in *Options: Preferences*, see page 59.

- *ASCII...:* Allows the import of experimental data in ASCII format with two columns containing channels and counts.

The data file format must be as follows:

- The file may contain an arbitrary number of comment lines at the beginning of the file. A comment line is a line that contains any non-numeric character. These lines will be ignored. The first line that contains only numeric characters will be treated as the first line of data.
- Each data line must consist of two columns: In the first column the channel number must be given (Integer), in the second column the number of counts must be given (Double). The two columns are separated by an arbitrary number of blanks or tabs.
- The decimal separator for floating point numbers must be '.', see also [subsection 3.2.5](#).
- Each line must end with <CR><LF>².
- The data file may contain up to 8192 channels.
- The data file may end with an arbitrary number of comment lines. A comment line is a line that contains any non-numeric character. These lines will be ignored.
- The name and path of the spectrum data file may contain arbitrary Unicode characters.
- The content of the data file has to be encoded in ANSI or UTF-8.

An example for a valid data file is given in [Figure 3.2](#).

- *ASCII (One column)...:* Allows the import of experimental data in ASCII format with one column containing counts.

The data file format is very similar to the *ASCII file format*, but contains the counts in one-column format without channel numbers. See [File: Read spectrum data: ASCII](#) for details.

²<CR> is the Carriage Return character (#13 decimal), <LF> is the Line Feed character (#10 decimal). Windows text editors add these characters automatically at the end of each line, and you don't have to care if the file originates from a Windows system. Data files from Unix systems may be problematic, because Unix uses a different end-of-line mark.

```

This line may contain a comment <CR><LF>
This line may contain a comment as well <CR><LF>
Channel  Counts <CR><LF>
1         1000  <CR><LF>
2         1000.0 <CR><LF>
3         1.0E3  <CR><LF>
4         +1.0E3 <CR><LF>
This line may contain an additional comment <CR><LF>
<EOF>

```

Figure 3.2.: Example for a valid data file which can be imported with *File: Read Data: ASCII...* The first three lines and the last line will be ignored by the program. The channel number must be an integer number, counts may be integer or floating point numbers.

```

This line may contain a comment <CR><LF>
This line may contain a comment as well <CR><LF>
Counts <CR><LF>
1000    <CR><LF>
1000.0  <CR><LF>
1.0E3   <CR><LF>
+1.0E3  <CR><LF>
This line may contain an additional comment <CR><LF>
<EOF>

```

Figure 3.3.: Example for a valid data file which can be imported with *File: Read Data: ASCII (One column)...* The first three lines and the last line will be ignored by the program. Counts may be integer or floating point numbers.

- The file may contain an arbitrary number of comment lines at the beginning of the file.
- Each data line consist of one column and contains the counts in the channel.
- Channels are zero based, i.e. the first line with counts belongs to channel zero.
- The decimal separator for floating point numbers must be '.'.
- Each line must end with <CR><LF>.
- The data file may contain up to 8192 channels.
- The data file may end with an arbitrary number of comment lines.
- The name and path of the spectrum data file may contain arbitrary Unicode characters.
- The content of the data file has to be encoded in ANSI or UTF-8.

An example for a valid data file is given in [Figure 3.3](#).

- *Canberra AVA...*: Allows the import of spectral data stored in Canberra's AVA file format. This file format is written for example by Canberra's ProSpect software. The AVA file

3. Using SIMNRA

format is an XML-based data format, see the *ProSpect Gamma Spectroscopy Software User's Manual* for details.

Note: SIMNRA reads spectral data stored in AVA files in the <spectrum> entry including real time, life time, filter parameters, pile-up rejector parameters, and acquisition start time. Other information stored in AVA files, such as energy calibration, list mode data etc., is ignored.

- *Canberra CAM...:* Allows the import of spectral data stored in Canberra's CAM file format.

SIMNRA uses Canberra's Genie-2000 software package for reading CAM files. The Genie-2000 package is not part of SIMNRA and must be obtained separately from Canberra Industries. This package must be installed correctly before you can read CAM files. The dynamic link libraries sad.dll etc. must be in the search path, and the virtual data manager (VDM) must be installed. SIMNRA has been tested with Genie-2000 versions from 1.4 to 3.4.1.

Note: SIMNRA reads only spectral data stored in CAM files including real time, life time, filter parameters and pile-up rejector parameters. Any other information which may be stored in the CAM file, like energy calibration etc., is ignored.

Note 2: Names of CAM files must contain ANSI characters only. Non-ANSI characters (i.e. arbitrary Unicode characters) are not allowed. This is a limitation of the Genie-2000 system, which does not support Unicode.

- *Read Spectrum Data: Energy/Counts...:* Allows the import of spectral data stored as energy – counts, with energy in keV.

The data points may be equidistant or non-equidistant in energy. If equidistant, then the data are used as they are. If the points are not equidistant they will be re-binned into equidistant channels. The new channel width can be given. In order to maintain correct counting statistics a stochastic element has to be introduced, so that each time when you read the same file the result will be different within count statistics.

The file format is very much like the [ASCII file format](#), see [page 12](#). The file may contain an arbitrary number of comment lines at the beginning of the file. A comment line is a line that contains any non-numeric character. These lines will be ignored. The first line that contains only numeric characters will be treated as the first line of data. Each data line must consist of two columns: In the first column the energy (in keV) must be given (Double), in the second column the number of counts must be given (Double). The two columns are separated by an arbitrary number of blanks or tabs. Each line must end with <CR><LF>³. The data points can be ordered with ascending or descending energy. The energy is the *lower* energy of the energy channel where the counts were recorded. Counts may be stored as floating point numbers in the file but must be real counts, i.e. whole

³<CR> is the Carriage Return character (#13 decimal), <LF> is the Line Feed character (#10 decimal). Windows text editors add these characters automatically at the end of each line, and you don't have to care if the file originates from a Windows system. Data files from Unix systems may cause trouble, because Unix uses a different end-of-line mark.

numbers without normalization. The data file may contain an arbitrary number of energy data points, but (after potential re-binning) the maximum number of channels is 8192.

The name and path of the spectrum data file may contain arbitrary Unicode characters, the content of the file has to be encoded in ANSI or UTF-8.

- *FAST MPA...*: Allows the import of spectral data stored in FAST ComTec's MPA file format. MPA files may contain more than one spectrum: In this case the spectrum for import can be selected.

Note: SIMNRA supports only MPA files with data in ASCII format. MPA files with data in binary, CSV or GANAAS formats are not supported.

Note 2: SIMNRA reads spectral data stored in MPA files including real time, life time and date/time of data acquisition. Any other information which may be stored in the MPA file, like energy calibration etc., is ignored.

- *IAEA SPE...*: Allows the import of spectral data stored in the SPE file format proposed by the International Atomic Energy Agency (IAEA).

Note: SIMNRA reads spectral data stored in SPE files including real time, life time and date/time of data acquisition. Any other information which may be stored in the SPE file, like energy calibration etc., is ignored.

- *IPP...*: Reads experimental data stored in the data file format used at the IPP Garching, Germany, until 1999. This data file format is outdated and will not be described here.
- *ISI...*: Reads experimental data stored in the data file format used at ISI, Jülich, Germany. This data file format will not be described here.
- *MCERD...*: Reads a spectrum calculated by the Monte-Carlo code MCERD, written by K. Arstila. The data are in the format:

Energy Counts, with energy in MeV. The data points must be equidistant in energy. The energy calibration is calculated automatically from the file.

- *RUMP spectrum...*: Reads an experimental spectrum from a RUMP RBS file. Additional information present in the file, such as incident particle, incident energy, geometry, etc. is ignored. The menu item [File: RUMP: Read RBS file](#) (section 3.3.7) allows reading all information stored in the file.
- *Swallow...*: This works much like RUMP's `swallow` command and allows to read experimental data stored in ASCII format without channel numbers. The command recognizes a simple file format, containing only data (see [Figure 3.4](#)), and an extended format which may contain additional comments at the beginning of the file. The extended format requires the key word `swallow`, see [Figure 3.5](#). Channel numbers start at 0 for compatibility with RUMP.

The name and path of the spectrum data file may contain arbitrary Unicode characters, the content of the file has to be encoded in ANSI or UTF-8.

3. Using SIMNRA

```
100 100 200 300 400 <CR><LF>
300 400 500 600 600 <CR><LF>
700 800 800 900 800 <CR><LF>
700 800 <CR><LF>
<EOF>
```

Figure 3.4.: Example for a simple ASCII file which can be imported with *File: Read Data: Swallow....* Data are counts in each channel. Lines may contain more than one channel. The last line should end with <CR><LF>. Empty lines at the top or bottom of the file are allowed. Channel numbers start with 0.

```
This line may contain any comment <CR><LF>
This line may contain any comment as well <CR><LF>
swallow<CR><LF>
100 100 200 300 400 <CR><LF>
300 400 500 600 600 <CR><LF>
700 800 800 900 800 <CR><LF>
700 800 <CR><LF>
<EOF>
```

Figure 3.5.: Example for an extended ASCII file which can be imported with *File: Read Data: Swallow....* The file may start with an arbitrary number of comment lines, which are ignored. The command 'swallow' must be written in its own line, followed by the data. Comment lines after the data are not allowed. Data are counts in each channel. Lines may contain more than one channel. The last line should end with <CR><LF>. Empty lines at the top or bottom of the file are allowed. Channel numbers start with 0.

The command *Read spectrum data:ASCCI (one column)* is more versatile in handling comment lines, but expects only one channel per line.

- *User...:* Allows to read experimental data stored in any user defined format. A dynamic link library (dll) has to be supplied by the user, which reads the data and passes them to SIMNRA. See [section 3.16](#) for more details.
- *XNRA and IDF...:* Allows the import of experimental spectrum data from an *xnra* or IDF file.

SIMNRA reads the following data from the file:

- the spectrum,
- real time and life time (if present),
- pulse shape parameters (if present),
- pile-up rejector parameters (if present).

Note: Alternatively a *xnra* or IDF file can be opened with *File: Open...* see subsection 3.3.2. If *File: Open...* is used, then SIMNRA will read the experimental setup, cross-section data, experimental and simulated spectra etc. from the file. *Read Spectrum Data: XNRA and IDF...* reads only the experimental spectrum. All other parameter remain unchanged.

3.3.6. Write Spectrum Data...

This menu item exports the experimental and simulated data as columns to an ASCII file. This file can be imported easily into any plot program, such as Excel, Origin or Mathematica. The file encoding is UTF-8.

The file format is as follows:

- The first line is a comment line containing information about the contents of the different columns.
- The content of the columns is as follows:
 - The first column is the channel number (if *Channel numbers* is checked).
 - The second column is the mean energy of each channel in keV (if *Energies* is checked).
 - The third column contains the experimental data (if *Experimental data* is checked). This column is set to zero if experimental data are selected but not available.
 - The fourth column contains smoothed experimental data (if *Smoothed experimental data* is checked).
 - The fifth column contains the simulated data (if *Simulated data (total)* is checked). This column is set to zero if simulated data are selected but not available.
 - The next columns contain simulated spectra for each element and isotope in the target, if *Simulated data (elements/isotopes)* is checked.
- Columns are separated by tabs.

3.3.7. RUMP

- *Read RBS File...* This menu item allows to read an RBS-file produced by the RUMP program containing experimental parameters (Type of incident particles, incident energy, scattering geometry, etc.) and spectral data.

If only the experimental spectrum without the experimental parameters should be read, then the menu item *File: Read spectrum data: RUMP spectrum* (section 3.3.5) has to be used.

The RBS-file may contain only one experimental spectrum. Compression level 3 (zero compression) is not implemented. See section 3.15 for more details.

RUMP stores the description of the sample and the absorber foil in sample description files (*.LCM). Sample description files can be read with *RUMP: Read Sample Description File...*

3. Using SIMNRA

- *Read Sample Description File...:* This menu item allows to read a sample description file produced by RUMP or the IBA data furnace NDF. The default file extension of sample description files is *.LCM.

RUMP stores the description of the sample and the absorber foil in sample description files (*.LCM). Experimental parameter (Type of incident particles, incident energy, scattering geometry, etc.) and spectral data are stored in files with extension *.RBS. RBS-files can be read with [RUMP: Read RBS File...](#)

SIMNRA supports only a subset of the RUMP sample description commands. The RUMP commands *Equation*, *Species* and *Fuzz* are not supported. If the sample description file contains these commands, they will be neglected and a warning will be shown. See [section 3.15](#) for more details.

- *Write Sample Description File...:* This menu item allows to store the structure of the target and the absorber foil in a sample description file in RUMP format. The default file extension is *.LCM. These files can be read by RUMP or the IBA data furnace NDF.

Note: SIMNRA allows to define a correction factor for the stopping power of each layer for each ion species. The correction factors are not stored in the sample description file.

3.3.8. Print...

This menu item will print all parameters of the calculation and plot the experimental and simulated curves. See also the print preferences on the [Options: Preferences](#) tab.

SIMNRA is not intended to produce high quality graphics. High quality graphics can be obtained by using a graphics program such as Excel or Origin. Data can be exchanged between SIMNRA and any graphics program via ASCII file with [File: Write Spectrum Data...](#) and via the clipboard with [Edit: Copy Data](#), see [section 3.15](#) for details.

3.3.9. Exit

Terminates the program.

3.4. **Edit menu**

3.4.1. **Copy data**

Copies experimental and simulated data in ASCII format to the clipboard, from where these can be pasted into any spreadsheet program.

The format of the data in the clipboard is as follows: The data are organised in three columns. The first column contains the channel number, the second column contains the experimental data, and the third column contains the simulated data. The columns are separated with tabs. Spectra of individual elements or isotopes are not copied to the clipboard. Use [File: Write Spectrum Data...](#) for exporting spectra of individual elements or isotopes to a file.

3.4.2. **Copy page**

Copies the visible graphics in enhanced metafile format to the clipboard. From there the graphics can be pasted into any presentation or word processing program such as Microsoft PowerPoint or Microsoft Word.

3.5. Setup menu

3.5.1. Experiment

In the *Setup: Experiment...* menu the global parameters of an experiment are defined. Less often used parameters (necessary for geometrical straggling and pile-up calculations, etc.) can be found in the *Setup: Experiment: More Options* menu.

- *Incident particle*: Selects the incident particle species. For incident neutrons (n), protons (H), deuterons (D), ^3He or ^4He ions the particles are selected by clicking the appropriate radio button.

For incident heavy ions select *Other* and enter the ions name in *Other particle: Element* (for example Si, Cl, I). Lowercase and uppercase letters in the ions name are treated similar, you can enter silicon as Si, si, SI or sI. The ions mass is selected from the drop down box.

Note: Because lowercase and uppercase letters are treated similar, both n or N for *Other particle: Element* will result in nitrogen as ion species. For selecting neutrons either the corresponding radio button can be clicked or *nn* can be used as input in *Other particle: Element*.

- *Energy*: Energy of the incident particles (in keV).
- *Geometry*: Geometry of the experiment: Incident angle α , exit angle β and scattering angle θ . α and β are measured towards the surface normal, see [Figure 3.6](#) and [Figure 3.17](#). All angles in degrees.

Note 1: $0^\circ \leq \alpha < 90^\circ$.

Note 2: $0^\circ \leq \beta \leq 180^\circ$. If $90^\circ < \beta \leq 180^\circ$ then transmission through the target is calculated.

Note 3: For ions $0.01^\circ \leq \theta \leq 180^\circ$. For neutrons $0^\circ \leq \theta \leq 180^\circ$.

Note 4: In most experimental setups either IBM or Cornell geometry is used. See [Figure 3.17](#) for a schematic representation of both geometries. You can use [Calculate: Exit Angle Beta...](#) to calculate β for IBM and Cornell geometries, see [subsection 3.8.9](#).

- *Calibration*: Conversion from channels to energy. To account for detector nonlinearities, SIMNRA can use a non-linear energy calibration with a quadratic term of the form

$$E [\text{keV}] = A + B \times \text{channel} + C \times \text{channel}^2. \quad (3.1)$$

E is the particle energy in keV. The calibration offset A must be entered in the *Calibration Offset* field, A in keV. The energy per channel B must be entered in the *Energy per Channel* field, B in keV/channel. C is the quadratic correction term, C in keV/channel². For a linear energy calibration $C = 0.0$. A linear calibration is appropriate in most cases, and only if a high accuracy is intended a non-linear calibration should be used.

The energy E according to [Equation 3.1](#) is the lower energy of a channel. The [Data Reader](#) displays the mean energy and lower/upper energy boundaries of a channel, see [subsection 3.9.1](#).

- *More energy calibration options*: An individual energy calibration may be used for each ion species. This is mainly useful for ERDA measurements with incident heavy ions, where each recoil species may require an individual calibration. By clicking the button an individual nonlinear energy calibration for each ion species may be supplied. If no individual energy calibration is defined, the major energy calibration entered in the *Calibration* fields of the *Setup: Experiment* form (see above) is used.

Attention: The energy scale, which is plotted at the top of the plot, is obtained with the major energy calibration from the *Calibration* fields of the *Setup: Experiment* form. The energy scale is not valid for particles with individual energy calibrations.

- *Particles*sr*: Number of incident particles times the solid angle of the detector. Solid angle in steradians. The number of incident particles is obtained from the collected charge and the charge state. [Calculate: Particles*sr...](#) can be used for calculating *Particles*sr*.
- *Detector Resolution*: Energy resolution of the detector (in keV). The energy resolution is measured as full width at half maximum (FWHM). This energy resolution is used for all ion species, if no specific resolution for that ion is supplied (see below). *Detector Resolution* is used only for solid state detectors. See [subsubsection 3.5.2](#), if you are using a different type of detector, such as a time-of-flight detector.

SIMNRA can use different detector energy resolutions for different ion species. By pressing the button detector energy resolutions for each ion species may be entered. If no energy resolution for an ion species is supplied, the default resolution (see above) is used.

Note 1: *Detector Resolution* is only available, if *Detector type* is *Solid state* in the [Setup: Experiment: More Options: Detector type...](#) menu. *Detector Resolution* is not used for other types of detectors, such as time-of-flight detectors.

Note 2: SIMNRA uses a constant, energy independent detector resolution for each ion species. This is (more or less) true for light ions (protons and He), but for heavy ions the detector resolution depends on the particle energy. Energy dependent solid-state detector resolutions are not yet implemented in SIMNRA.

- *Energy spread of incident beam*: Usually the incident ion beam is not monoenergetic but has an energy distribution. SIMNRA can use a Gaussian or an asymmetric Gaussian energy distribution of the incident beam. If *Energy spread of incident beam* is set to 0.0 a monoenergetic incident beam is used. For a Gaussian energy distribution the full width at half maximum (FWHM) has to be entered in the *Energy spread of incident beam* field. By pressing the button the parameters for an asymmetric Gaussian energy distribution can be entered. An asymmetric Gaussian distribution is a two-piece normal distribution (TPND), with

$$f(E) = \begin{cases} A \exp\left(-\frac{(E-E_0)^2}{2\sigma_{\text{Low}}^2}\right) & \text{if } E \leq E_0 \\ A \exp\left(-\frac{(E-E_0)^2}{2\sigma_{\text{High}}^2}\right) & \text{if } E > E_0 \end{cases} \quad (3.2)$$

E_0 is the energy at which the distribution has its maximum (called mode in statistics), σ_{Low} and σ_{High} are the standard deviations towards lower and higher energies, and A is

3. Using SIMNRA

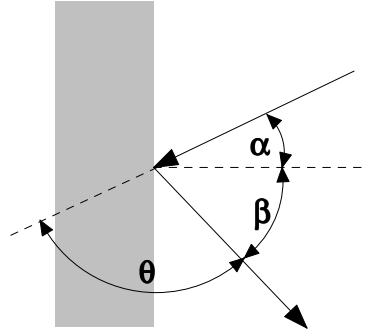


Figure 3.6.: Geometry of a scattering experiment. Incident angle α , exit angle β and scattering angle θ .

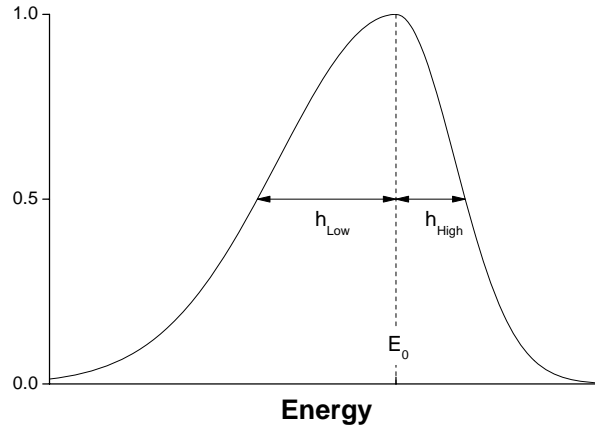


Figure 3.7.: Parameters of an asymmetric Gaussian energy distribution. E_0 is the energy at which the distribution has its maximum, h_{Low} is the half width at half maximum (HWHM) towards lower energies, and h_{High} the HWHM towards higher energies. The HWHM is the distance between E_0 and the energy, at which the distribution has reached half of the maximum value. h_{Low} and h_{High} are always positive numbers.

a scaling factor. SIMNRA characterises asymmetric Gaussian distributions by their half widths at half maximum (HWHMs) towards lower and higher energies, see [Figure 3.7](#). HWHM h and standard deviation σ are connected through

$$h = \sqrt{2 \ln 2} \sigma. \quad (3.3)$$

The FWHM of an asymmetric Gaussian distribution is simply the sum of h_{Low} and h_{High} . A Origin fitting function for asymmetric Gaussian distributions can be found in [Appendix E](#).

Note: Asymmetric Gaussian energy distributions are only available if *Shape of straggling distributions* is set to *Asymmetric Gaussian* in the *Setup: Calculation: Parameter* menu, see subsection 3.5.3.

The *File*-menu allows to save experimental setups to disk and read experimental setups from disk. All information contained in the *Setup Experiment* form, the *Detector geometry* form, all energy calibrations and all detector resolutions are saved. The *xnra* data file format is used for saving, the default file extension is *xsetup*.

- *Save as Default*: The current experimental setup is stored as startup default for SIMNRA in the file *setup.xsetup*, see [Table 2.1](#). This affects all users of the computer.
- *Save Setup as*: Save the current experimental setup to file.
- *Read Setup*: Reads an experimental setup from file.
Note: You can read any *nra*-, *xnra*-, or *IDF*-file with *Read Setup*. Only the experimental setup information will be read, any other information (such as target composition, experimental spectra etc.) which may be present in the file, will be ignored.

3. Using SIMNRA

3.5.2. Experiment: More Options

Detector type...

The information in this menu is only necessary, if a special detector is used. Special detectors are

1. thin solid state detectors. A solid state detector is thin, if particles are not fully stopped in the detector, but lose only a fraction of their energy. This can be the case for thin transmission detectors (with thicknesses of 10 μm or below), or for nuclear reactions creating high energetic protons: The penetration depth of 10 MeV protons in silicon is about 700 μm , so that these protons are only partly stopped in typical silicon detectors with thicknesses of the order of 100 μm ,
 2. time-of-flight detectors,
 3. electrostatic detectors. These are often used in medium ion scattering (MEIS).
- *Detector type*: Select the type of detector: Either *solid state* (SSD), *time-of-flight* (TOF), or *electrostatic* detector.
 - *Solid state detector*:
 - *Detector thickness*: Thickness of the SSD detector, in μm . Select *Infinity*, if the particles are fully stopped in the detector.
 - *Material*: Detector material. Silicon detectors are assumed. This cannot be changed.
 - *Time-of-flight detector*:
 - *Free flight path*: Length of the flight path, for which the time of flight is measured, in m.
 - *Time resolution*: Time resolution of the TOF detector, in ps. The full width at half maximum (FWHM) has to be used. The time resolution is used to calculate the energy resolution of the TOF detector, see [subsection 4.7.1](#).
 - *Electrostatic detector*:
 - *Delta-E/E*: The energy resolution of electrostatic detectors is usually given by a constant ratio $\Delta E/E$, with ΔE the energy resolution in FWHM and E the particle energy. See also [subsection 4.7.2](#).
 - *Detector sensitivity*: Allows to supply a detector sensitivity, i.e. an energy-dependent detection probability.
 - *Unity*: The detector sensitivity is equal to one at each energy, i.e. each particle is detected.
 - *From file*: Allows to specify a file containing the detector sensitivity. The file can be an ASCII data file, an *xnra-file*, or an IDF-file.

```

This line may contain any comment <CR><LF>
This line may contain any comment as well <CR><LF>
Energy (keV)   Sensitivity <CR><LF>
100.0          0.1 <CR><LF>
200.0          0.3 <CR><LF>
300.0          0.8 <CR><LF>
400.0          1.0 <CR><LF>
<EOF>

```

Figure 3.8.: Example for a valid ASCII data file which can be imported with *Detector sensitivity: From File*. The first three lines will be ignored by the program.

The format of the ASCII data file is very much like the ASCII file format for spectra, see [section 3.3.5](#). The file may contain an arbitrary number of comment lines at the beginning of the file. A comment line is a line that contains any non-numeric character. These lines will be ignored. The first line that contains only numeric characters will be treated as the first line of data. Each data line must consist of two columns: In the first column the energy (in keV) must be given (Double), in the second column the detector sensitivity as a number between 0.0 and 1.0 must be given (Double). The two columns are separated by an arbitrary number of blanks or tabs. Each line must end with <CR><LF>⁴. The data points must be ordered with ascending energy. The maximum number of data points is 1000. Linear interpolation is used between the given data points. At energies below the lowest given energy value and above the highest energy value the sensitivity is assumed to be unity. An example for a valid file is shown in [Figure 3.8](#).

The name and path of the spectrum data file may contain arbitrary Unicode characters, the content of the file has to be encoded in ANSI or UTF-8.

Detector Geometry...

In this menu the detailed geometry of the detector (beam diameter, detector diaphragm width, distance sample-detector and shapes of incident beam and detector diaphragm) is entered. This is only necessary if geometrical straggling due to finite widths of the incident beam and detector diaphragm should be calculated. Geometrical straggling is usually small for RBS, but may be considerable for ERDA and for some cases of NRA. See [Figure 3.9](#) for details how distances and diameters are measured.

- *Calculate geometrical straggling*: Check to include geometrical straggling in the simulation. Geometrical straggling is neglected if his box is unchecked.

⁴<CR> is the Carriage Return character (#13 decimal), <LF> is the Line Feed character (#10 decimal). Windows text editors add these characters automatically at the end of each line, and you don't have to care if the file originates from a Windows system. Data files from Unix systems may cause trouble, because Unix uses a different end-of-line mark.

3. Using SIMNRA

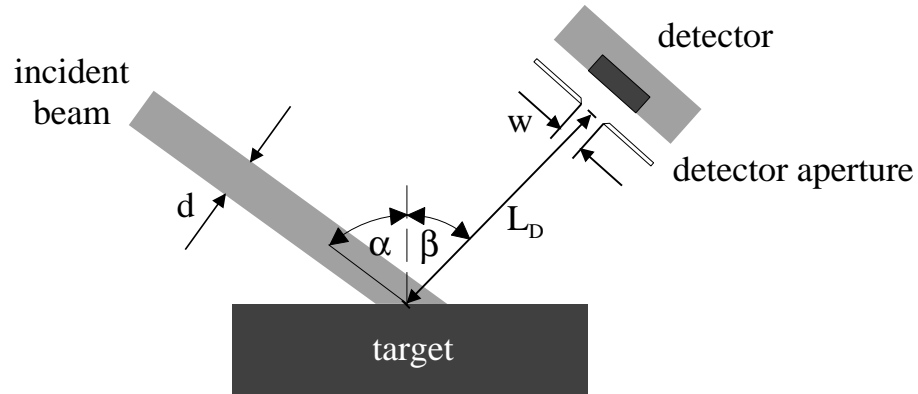


Figure 3.9.: Detector geometry. d is the diameter of the incident beam, w the width of the detector aperture and L_D the distance between sample and the detector aperture. Incident angle α and exit angle β .

- *Incident beam:* Geometrical parameters of the incident beam.
 - *Shape of incident beam:* Circular or Rectangular beams can be selected. A homogeneous current distribution of the incident beam is assumed.
 - *Diameter of incident beam:* For circular beams: Diameter of the incident beam in mm. Note that the size of the beam spot on the sample surface is $d/\cos \alpha$, see Figure 3.9.
 - *Width and Height of incident beam:* For rectangular beams: Width and height of the incident beam in mm.
- *Detector position:* Position of the detector. See Figure 3.17 for IBM and Cornell geometries. In many cases SIMNRA will determine the detector position automatically. In these cases *Detector position* is grayed out. In some cases, however, automatic determination of the detector position is not possible due to geometrical ambiguities. This is, for example, the case for normal incidence.
- *Detector aperture:* Geometrical parameters of the aperture in front of the detector.
 - *Shape of detector aperture:* Circular or Rectangular apertures can be selected.
 - *Diameter of detector aperture:* For circular apertures: Diameter of the detector aperture in mm.
 - *Width and Height of detector aperture:* For rectangular apertures: Width and height of the detector aperture in mm.
- *Distance sample-detector aperture:* Distance between the sample surface and the detector aperture in mm.

Note: *Straggling* in the *Setup: Calculation* menu must be checked in order to calculate geometrical straggling (and electronic energy loss straggling).

Live-time and pile-up...

The parameter for live-time correction and pile-up simulation are entered in this menu. If experimental data are read from [Canberra's CAM-files](#) then some of these parameters are taken from the information stored in the files.

- *Apply live-time correction*: Check if a live-time correction should be applied to simulated spectra. After checking *Apply live-time correction* the values for real and live time have to be entered. Calculation of pile-up is only possible if *Apply live-time correction* is checked.
- *Real time*: Real time of the measurement (in s). The real time is the time it took to measure the spectrum. The real time is necessary for live-time corrections and pile-up calculations. If Canberra's CAM-files are used, this parameter is taken from the file.
- *Live time*: Live time of the measurement (in s). The live time is the time interval during the measurement the analog-digital converter (ADC) was able to accept pulses. This value can be obtained from your multi-channel analyzer (MCA). The live time is almost identical to the real time for low pulse rates. The live-time is only necessary for live-time corrections and can be set identical to the real time, if a pile-up calculation is desired without knowing the correct live-time.

If Canberra's CAM-files are used, this parameter is taken from the file.

- *Calculate pile-up*: Check if pulse pile-up should be calculated for simulated spectra. After checking *Calculate pile-up* you have to enter a value for the pulse rise-time or the fudge time parameter. *Calculate pile-up* is only available if *Apply live-time correction* is checked.

Note: The pile-up calculation gives incorrect results for ADC offsets other than zero.

- *Pulse rise time*: Rise time of the amplified pulse from zero to its maximum value (in μs). See [Figure 3.10](#) for a schematic definition of the pulse rise time. For Gaussian pulse shaping, the pulse rise time T_p can be derived from the Gaussian shaping time τ_{ST} from [44]

$$T_p = 2.2 \times \tau_{ST}. \quad (3.4)$$

The pulse rise time can be measured accurately with an oscilloscope at the amplifier output. However, as SIMNRA approximates the true pulse shape with a parabola (see [subsection 4.13.2](#), [Figure 4.44](#) and [Figure 4.45](#)), it is advantageous to use a slightly smaller value for T_p than obtained from [Equation 3.4](#), see [Figure 3.11](#) and [Table 3.1](#).

If [Canberra's CAM-files](#) are used, then the pulse rise time is computed from the values for the Gaussian shaping time, DSP rise time, and DSP flat top duration stored in the file according to [Table 3.1](#).

Note: *Pulse rise time* is only available, if the *pile-up model* is set to *Accurate* in the *Setup:Calculation* menu, see [subsection 3.5.3](#).

- *Fudge time parameter*: Fudge time parameter for the *Fast pile-up model* (in μs). This parameter can be adjusted to any value matching the measured pile-up in the spectrum. Reasonable values are in the range 0.3–0.5 μs , see [subsection 4.13.2](#) for more details.

3. Using SIMNRA

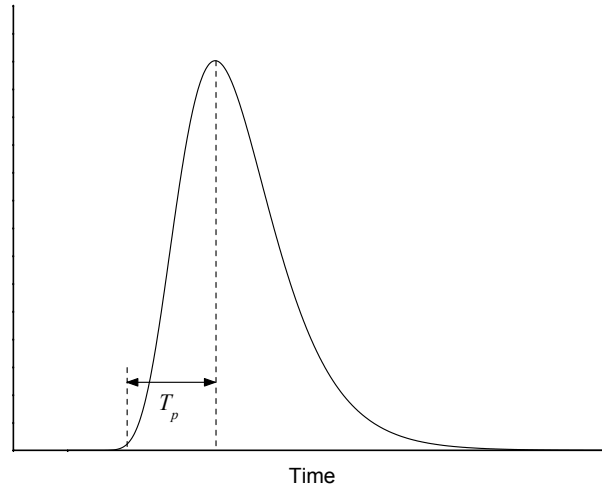


Figure 3.10.: Schematic representation of an amplified pulse with Gaussian shaping. T_p is the pulse rise time.

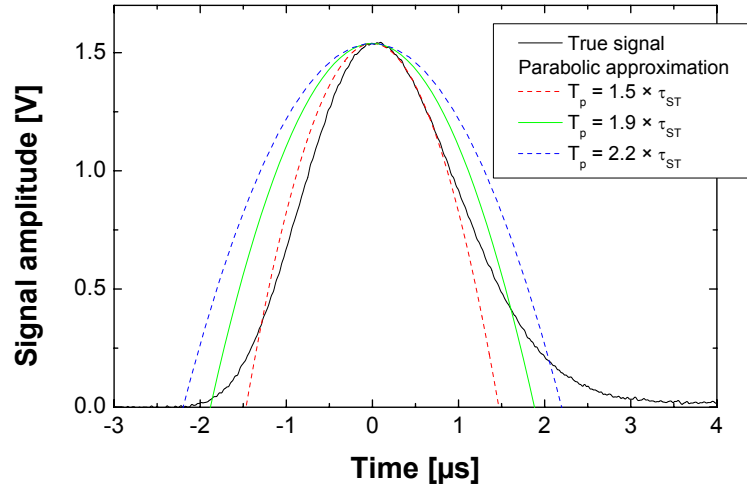


Figure 3.11.: Selection of the pulse rise time T_p for a Gaussian pulse. True pulse from an Ortec 672 spectroscopy amplifier, shaping time $\tau_{ST} = 1 \mu s$. SIMNRA approximates the Gaussian pulse shape with a parabola. The figure shows parabolas with rise times T_p equal to 1.5, 1.9, and 2.2 times the Gaussian shaping time. The parabola with $T_p = 1.9 \times \tau_{ST}$ usually gives the best approximation to the true pulse shape.

Table 3.1.: Recommended values for the pulse rise time T_p .

Device	Pulse rise time T_p	
Amplifier (Gaussian shaping)	$T_p = 1.9 \times \tau_{ST}$	τ_{ST} : Gaussian shaping time
Digital signal processor (DSP)	$T_p = 0.8 \times (\tau_{RT} + 0.5 \times \tau_{FT})$	τ_{RT} : DSP filter rise time τ_{FT} : DSP filter flat top duration

Note: The *Fudge time parameter* is only available, if the *pile-up model* is set to *Fast* in the *Setup:Calculation* menu, see subsection 3.5.3.

- *Pile-up rejector*: Switch to *On*, if a pile-up rejector was used during the measurement. Switch to *Off*, if the measurement was made without a pile-up rejector.

If Canberra's CAM-files are used, this parameter is taken from the file. But note that this is not always reliable, because it cannot be excluded that the pile-up rejector was enabled, but had no influence due to incorrect or missing cabling.

Note: This switch is only available if the *pile-up model* is set to *Accurate* in the *Setup:Calculation* menu. This switch has no influence, if the *pile-up model* is set to *Fast*.

- *Pile-up rejector pair-resolution time*: Pair resolution time of the pile-up rejector (in μs). A pile-up rejector (PUR) is only able to recognize two pulses as different, if their time difference is larger than the pair-resolution time. The pair-resolution time is normally specified in the technical manual of the pile-up rejector. Typical values are in the range 0.3–0.5 μs .

Note: This parameter is only available, if the pile-up model is set to *Accurate* in the *Setup:Calculation* menu. This parameter has no influence, if the pile-up model is set to *Fast*.

3. Using SIMNRA

3.5.3. Calculation

In the *Setup: Calculation...* menu the calculation parameters can be altered. This affects the accuracy of the calculation, but also the time necessary to calculate a simulated spectrum.

Parameter tab

- *Isotopes*: If checked, scattering from all isotopes of all elements in the target is calculated. Especially for heavy elements with many isotopes this will slow down the calculation significantly. If unchecked, the program will use only the mean masses of the elements. Default is checked.

Important: Non-Rutherford cross-sections and nuclear reactions are only available if *Isotopes* is checked.

- *Straggling*: If checked, then energy-loss straggling in window, target and foil is taken into account. Geometrical straggling can be taken additionally into account by selecting geometrical straggling, see [subsubsection 3.5.2](#). Energy spread by small-angle multiple scattering can be taken into account by selecting multiple scattering, see [section 3.5.3](#).

If *Straggling* is not checked, then all energy straggling components (energy-loss straggling, geometrical straggling, energy straggling by multiple scattering) are neglected independent of the settings of geometrical straggling in [subsubsection 3.5.2](#) and multiple scattering in [section 3.5.3](#). Default is checked.

- *Multiple scattering*: If checked, straggling due to multiple small angle scattering in window, target and foil will be calculated. Default is unchecked.

The model used for multiple-scattering calculations can be selected with [Multiple-scattering model](#).

Multiple scattering increases the accuracy of simulations and should be checked if the target contains heavy elements and/or thick layers. However, the implemented multiple scattering theory fails for large energy losses, and the low-energy part of the simulated spectrum gets unreliable if *Multiple Scattering* is checked.

- *Dual Scattering*: Most particles are scattered into the detector with only one scattering event with large scattering angle. However, some particles may suffer more than one scattering event with large scattering angle before they reach the detector, see [Figure 3.12](#). This is called plural scattering and results for example in the low background behind the low energy edge of high Z layers on top of low Z elements. The deviations at low energies between simulated and measured spectra are also mainly due to plural scattering.

SIMNRA can calculate all trajectories with two scattering events. If *Dual Scattering* is unchecked, then only one scattering event is calculated. This is the default. If *Dual Scattering* is checked, additionally trajectories with two scattering events will be calculated.

Warning: The calculation of dual scattering is a time consuming process. If *Dual Scattering* is checked, this will slow down the calculation of a spectrum by a factor of about 200 (!).

Note 1: If non-Rutherford cross-sections are selected for some elements, then the selected non-Rutherford cross-sections will be used for the single scattering calculation. The dual scattering calculation, however, is always performed with Rutherford cross-sections: Dual scattering requires cross-sections at all possible scattering angles between $\approx 0^\circ$ and 180° , which are only available in the case of Rutherford cross-sections. This is reasonable in many cases, because dual scattering is usually dominated by heavy elements in the target, where the cross-sections are Rutherford up to high energies. However, this approximation may result in incorrect spectra, if non-Rutherford scattering from light elements is important. SIMNRA will issue a warning message if dual scattering is used with non-Rutherford cross-sections.

Note 2: SIMNRA calculates dual scattering *only* for incident ions and not for recoils or reaction products of nuclear reactions. Additional scattering in a foil in front of the detector (if any) is neglected.

Note 3: If *Dual Scattering* is checked, then *Straggling* must be checked too. SIMNRA will check *Straggling* automatically, if *Dual Scattering* is checked. As long as *Dual Scattering* is checked, *Straggling* cannot be unchecked.

- *Electronic stopping power data:* The selection of electronic stopping power data has a large influence on the shape of simulated spectra. SIMNRA can use different sets of electronic stopping power data for the stopping of light and heavy ions in all elements:
 - *Andersen/Ziegler:* Electronic stopping power data by Andersen and Ziegler [6, 131, 136].

Note: If Andersen/Ziegler stopping power data are used for incident hydrogen isotopes or heavy ions near 1 MeV/amu, artificial steps or kinks may appear in the simulated spectra. This is due to a jump of the stopping power at 1 MeV/amu. See the description of the *High energy stopping* switch for a work-around.
 - *Ziegler/Biersack:* Electronic stopping power data by Ziegler, Biersack and Littmark [136]. These data are identical to Ziegler's SRIM 1997 (formerly TRIM) program. The Ziegler-Biersack data are generally more accurate and reliable than the Andersen-Ziegler data [133]. The energy ranges in which the different stopping power formulas are valid are listed in Table 3.2.
 - *ZB+KKK:* Identical to *Ziegler/Biersack* for most ion/target combinations, except for H, D, T, ^3He and ^4He in C and Si, in which case the stopping power data by Konac et al. [61, 62] are used. The KKK stopping powers are valid in the energy range $0.01 \leq E \leq 100$ MeV/amu. See subsection 4.6.3 for details about the KKK stopping powers.
 - *SRIM:* Stopping power data from Ziegler's SRIM 2003 or higher program. This option will work only, if
 1. SRIM 2003 or higher is installed on your computer,
 2. the path to the SRIM directory is registered correctly in *Options: Preferences: Directories*.

3. Using SIMNRA

See [section 3.18](#) for more details.

- *DPASS*: Stopping power data from Schinner’s and Sigmund’s DPASS program [94, 95, 100, 101]. DPASS stopping powers are available for all ions in all targets in the energy range 0.001 to 1000 MeV/amu.
- *User defined*: User defined stopping powers are used. See [section 3.19](#) for details. But note that stopping power data for all elements present in the target have to be supplied if *User defined* stopping powers are selected.
- *High energy stopping*: This switch is only available if [Andersen-Ziegler](#) electronic stopping power data are used. If checked, the program will use the correct high energy stopping formula by Andersen and Ziegler for incident protons and heavy ions for $E > 1$ MeV/amu. If unchecked, the program will use the medium energy formula, which is valid only in the range 10 keV/amu–1 MeV/amu, also at higher energies $E > 1$ MeV/amu. The difference between the two formulas usually is small. This switch is necessary because the two stopping power formulas do not fit smoothly together at 1 MeV/amu: The stopping power jumps at 1 MeV/amu resulting in artificial kinks and steps in the simulated spectra. This problem is overcome if *High energy stopping* is unchecked, however this will result in less accurate stopping powers for energies > 1 MeV/amu. A better solution is to use the Ziegler-Biersack stopping powers, where this problem does not occur.

This switch does not have any influence on the calculation of the stopping power of helium ions and is disabled. For helium ions always the medium energy formula is used, which is valid for all energies below 10 MeV. The default is checked.

- *Nuclear stopping power data*: Selects the nuclear stopping power model. The nuclear stopping power is usually much smaller than the electronic stopping power for incident protons and helium ions, but it may have a strong effect for heavy ions. Either *None* for no nuclear stopping or *Universal* for nuclear stopping calculated with the universal potential [136] (see [subsection 4.6.5](#)) can be selected. In almost all cases *Universal* should be selected, which is the program default.
- *Shape of straggling distributions*: Selects which shape is used for all straggling distributions. *Gaussian* shape and *Asymmetric Gaussian* shape with a two-piece normal distribution can be selected. This selection influences the energy distributions by energy-loss straggling, geometrical straggling and multiple scattering. *Asymmetric Gaussian* is more accurate and the program default.
- *Energy-loss straggling model*: Selects the electronic energy-loss straggling model. Bohr’s theory, Chu’s theory, or Chu + Yang’s theory can be used. Chu’s theory takes deviations from Bohr straggling caused by the electron binding in the target atoms into account, and Yang’s theory additionally incorporates charge state fluctuations of the ions. See [subsection 4.8.2](#) for more details about electronic energy-loss straggling. Chu + Yang’s theory is recommended.
- *Multiple-scattering model*: Selects the multiple small-angle scattering model. The models developed by E. Szilagy et al. [109] or by G. Amsel et al. [2] using Thomas-Fermi

screening can be selected. SIMNRA 7.02 and earlier always used the *Szilagyi* model. Both models are usually very close together, with the *Amsel* model being slightly more accurate. The *Amsel* model is recommended.

Note: *Multiple-scattering model* only has an effect if *Multiple scattering* is checked.

- *Weighting of cross-sections by straggling*: Selects how the energy distribution by straggling is included in the calculation of the cross-section. See [section 4.4](#) for details.
 - *Fast* model: This model does not include straggling in the calculation of the cross-section. It computes faster and is sufficiently accurate for cross-sections varying slowly with energy, such as the Rutherford cross-section. It may be, however, slightly inaccurate if the cross-section contains sharp structure. The *Fast* model was used by SIMNRA 6 and earlier.
 - *Accurate* model: This model uses a mean cross-section weighted with the energy straggling distribution and takes the influence of the shape of the cross-section on the moments of the energy distribution of outgoing particles into account. It is generally more accurate than the *Fast* model, especially if the cross-section contains sharp structures, but requires more calculation time. The *Accurate* model is recommended and the program default.
- *Screening to Rutherford cross-section*: Selects the screening function to the Rutherford cross-section due to partial screening of the nuclear charges by the electron shells surrounding both nuclei, see [section 4.3](#).
 - *None*: Rutherford cross-section without screening, see [Equation 4.11](#). This option should be used only for test purposes, because the unscreened Rutherford cross-section may deviate considerably from the correct one, especially at low energies, small scattering angles and high-Z elements.
 - *L'Ecuyer*: Screening function according to L'Ecuyer, see [Equation 4.13](#). The L'Ecuyer screening function is only reasonable for RBS at sufficiently high energies and backscattering angles $\theta > 90^\circ$. It should be used only for test purposes due to limited accuracy at low energies and small scattering angles. The *Andersen* screening function is generally a better choice.

Note: If L'Ecuyer screening is selected for recoils, this selection is ignored and replaced by Andersen screening.

 - *Andersen*: Screening function according to Andersen, see [Equation 4.14](#). Andersen's screening function provides high accuracy for typical ion beam analysis applications, i.e. if the energy is above several hundred keV. Andersen screening is the program default. Andersen's screening function was developed for weakly screened potentials, so that Andersen screening may become inaccurate at very low energies and/or small scattering angles. In these cases *Universal* screening provides higher accuracy.
 - *Universal*: Universal screening function, developed by Ziegler *et al.* [136]. The Universal screening function was developed for strongly screened potentials and

3. Using SIMNRA

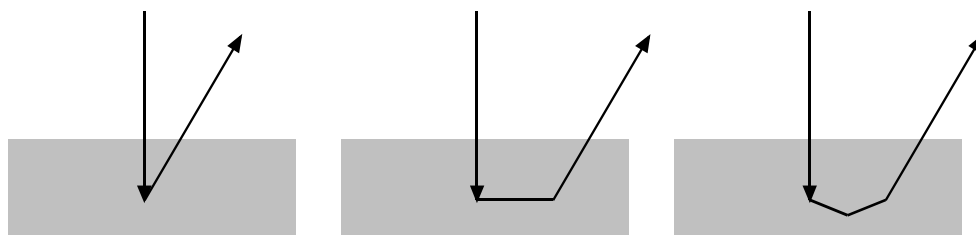


Figure 3.12.: Examples of ion trajectories with one, two and three scattering events.

Table 3.2.: Energy ranges in which the different stopping power formulas are valid.

Incident ion	Andersen-Ziegler (keV/amu)	Ziegler-Biersack (keV/amu)
Hydrogen (H, D, T)	1–100000	1–100000
Helium (^3He , ^4He)	0.25–2500 ^a	1–100000
Heavy ions	1–100000	1–100000

^aHigh energy electronic stopping formula for energies > 2.5 MeV/amu from [6] not implemented.

provides high accuracy even at low energies and small scattering angles. At higher energies and larger scattering angles it becomes very similar to the Andersen screening function. Universal screening should be selected for small incident energies and/or small scattering angles.

Note: Because Universal screening approaches Andersen screening at higher energies, Universal screening is replaced by Andersen screening at energies above *CrossSec.UniversalEMax* (see [page 322](#)).

Options tab

- *Logfile*: If checked, a file named `SIMNRA.LOG` is created. This file contains additional information about each step of the calculation and can be viewed with the program [VIEWNRA](#). The log-file is written to the current directory, if an `nra-` or `xnra-file` was opened or saved. Otherwise it is written to the temporary directory defined by the operating system. Default is unchecked.

Stepwidths tab

- *Stepwidth incident ions*: Stepwidth of the incident ions used in the calculation. See [chapter 4](#) for details. *Automatic* or *Fixed* can be selected. If *Automatic* is selected, the program will choose the stepwidth automatically. This is usually the best choice for obtaining high accuracy and small computing times. *Automatic* is the program default. The automatically determined stepwidth is kept always below the resolution of the experiment. Because

the resolution in larger depths degrades due to energy loss straggling the program uses a small stepwidth near the surface and a larger stepwidth in larger depths.

Fixed stepwidth: The program uses a fixed stepwidth for the calculation. If *Fixed* is selected the default for the stepwidth is 10 keV. For incident heavy ions with energies in the range of several ten MeV this stepwidth can be increased to several 100 keV.

The stepwidth of incident ions affects the time T necessary to perform a calculation strongly. T depends on the stepwidth of the incoming ion ΔE roughly as $T \propto 1/\Delta E$: Decreasing the stepwidth by a factor of two will roughly double the computing time.

Note 1: The stepwidth of incident ions is an important parameter for the accuracy of a simulation. The accuracy may be increased, if a small *Fixed* stepwidth is used instead of *Automatic* step width control, but usually at the cost of higher computing times.

Note 2: Energy-loss and straggling calculations try to keep the numerical inaccuracy below some limits, see [Accuracy of energy-loss calculation](#) on page 36 and [Accuracy of straggling width calculation](#) on page 36. If the *Fixed* stepwidth is too large and the numerical inaccuracy exceeds these limits, then the internal step width of energy-loss and straggling routines is decreased automatically. The internal step width of energy-loss and straggling routines therefore will never exceed the *Fixed* value, but may be nevertheless lower. A large *Fixed* stepwidth therefore can speed up calculations in some cases, but may have only limited use for speeding-up simulations in other cases.

Note 3: If the scattering cross-section contains narrow resonances and [Weighting of cross-sections by straggling](#) is set to *Fast* (see [subsubsection 3.5.3](#)), a *Fixed* stepwidth of incident ions (with a step width smaller than the width of the resonances) gives best results. If *Automatic* stepwidth control is used together with narrow resonances, then [Weighting of cross-sections by straggling](#) should be set to *Accurate*.

Note 4: If a *Fixed* stepwidth of incident ions is used and the stepwidth is too high, unwanted oscillations or steps in the simulated spectra may occur. This is due to rounding errors in the routine which calculates the contents of each channel. If these oscillations occur, you have to decrease the stepwidth of incident ions. These problems should never occur with *Automatic* stepwidth control: The program always uses a stepwidth which is small enough.

- *Stepwidth outgoing ions*: Stepwidth of outgoing particles used in the calculation. See [chapter 4](#) for details. *Automatic* or *Fixed* can be selected. If *automatic* is selected, the program will choose the stepwidth automatically. This is usually the best choice for obtaining high accuracy and small computing times. *Automatic* is the program default. The automatically determined stepwidth is large at high energies, where the stopping power shows only small variations to decrease computing time. The step width is decreased near the stopping power maximum and at low energies, where the stopping power varies strongly, to increase accuracy.

Fixed stepwidth: The program uses a fixed stepwidth for the calculation of outgoing particles. The stepwidth will remain constant at all energies and for all outgoing particles. The internal step width of energy-loss and straggling routines may be decreased below

3. Using SIMNRA

the *Fixed* value (if necessary, see the [Note](#) below), but will never exceed the *Fixed* value. If *Fixed* is selected the default for the stepwidth is 200 keV. For incident heavy ions with energies in the range of several ten MeV this stepwidth can be increased.

If a small *Fixed* stepwidth is used this may increase the accuracy of the calculation, but will slow down the calculation. A very small *Fixed* stepwidth may be more accurate than *Automatic* stepwidth control. In contrast a large *Fixed* stepwidth usually decreases the accuracy of the calculation, but will speed up the calculation to some extent. There is no easy recipe for the best choice of a *Fixed* stepwidth. Usually the best compromise between speed and accuracy is *Automatic* stepwidth control.

Note: Energy-loss and straggling calculations try to keep the numerical inaccuracy below some limits, see [Accuracy of energy-loss calculation](#) on page 36 and [Accuracy of straggling width calculation](#) on page 36. If the *Fixed* stepwidth is too large and the numerical inaccuracy exceeds these limits, then the internal step width of energy-loss and straggling routines is decreased automatically. The internal step width of energy-loss and straggling routines therefore will never exceed the *Fixed* value, but may be nevertheless lower. A large *Fixed* stepwidth therefore can speed up calculations in some cases, but may have only limited use for speeding-up simulations in other cases.

- *Cutoff Energy:* All particles are calculated until their energy has decreased below the cut-off energy. You may speed up the calculation if you increase the cut-off energy. The lowest possible value for the cut-off energy is 1 keV, the default is 10 keV.

Note: The cut-off energy is only used for charged particles, not for neutrons.

- *Accuracy of energy-loss calculation:* Accuracy of the energy-loss calculation, in keV per Runge-Kutta step. The internal step width of the energy-loss calculation is chosen small enough, so that the estimated numerical inaccuracy during one step is smaller than the *Accuracy of energy-loss calculation*. Inaccuracies of multiple steps are roughly independent and add quadratically. See [section 4.5](#) for details. The estimated inaccuracies of energy-loss calculations can be found in the *Logfile*, see [subsubsection 3.5.3](#).

Accuracy of energy-loss calculation cannot be changed.

- *Accuracy of straggling width calculation:* Accuracy of the the straggling width calculation, in % of the full width at half maximum (FWHM) of the straggling per step. This accuracy is used for electronic energy-loss straggling and straggling induced by sample porosity. The internal step width of straggling calculations the energy-loss calculation is chosen small enough, so that the estimated numerical inaccuracy during one step is smaller than the *Accuracy of straggling width calculation*. Inaccuracies of multiple steps are not independent but add about linearly for the variances. The estimated inaccuracies of straggling calculations can be found in the *Logfile*, see [subsubsection 3.5.3](#).

Accuracy of straggling width calculation cannot be changed.

Pile-up tab

- *Pile-up model*: Selects the model for pile-up calculations, see [subsection 4.13.2](#). The *Fast* model is less accurate, but is calculated fast. The *Accurate* model is closer to physical reality, but the calculation takes much longer. The computing time is about $\propto N^3$ (with N the number of channels in the spectrum) for the *Accurate* model, but only $\propto N^2$ for the *Fast* model: For 1000 channels the *Fast* model is more than 1000 times faster.

The *Fast* model can be applied, if a pile-up rejector is used and the pulse rise time is larger than about $1\ \mu\text{s}$. The *Accurate* model should be used, if the measurement was done without a pile-up rejector, or if fast pulses are used, i.e. if the pulse rise time is smaller than $1\ \mu\text{s}$. See [subsection 4.13.2](#) (especially [Figure 4.47](#)) for more details.

Default is the *Fast* model.

Note: Several parameters in the *Setup: Experiment: Live time and pile-up* form are only available if the *Pile-up model* is set to *Accurate*, see [subsubsection 3.5.2](#).

Roughness tab

- *Number of thickness steps*: Used for the calculation of layer roughness. A rough layer is approximated by the superposition of N spectra with different layer thicknesses, where N is the number of thickness steps. If N is small, the superposed spectrum may contain steps. Larger values of N result in smoother spectra, but slow down the calculation considerably. Default is $N = 10$.

Note: One rough layer requires the calculation of N spectra, two rough layers of N^2 spectra, three rough layers of N^3 spectra etc.

- *Number of angular steps*: Used for the calculation of substrate roughness. A rough substrate is approximated by the superposition of M spectra with different incident and exit angles, where M is the number of angular steps. If M is small, the superposed spectrum may contain steps. Larger values of M result in smoother spectra, but slow down the calculation considerably. Default is $M = 20$.

Note: Substrate roughness requires the calculation of M spectra, one rough layers combined with substrate roughness of $N \times M$ spectra, where N is the [Number of thickness steps](#), two rough layers combined with substrate roughness of $N^2 \times M$ spectra etc.

- *Dimension of substrate roughness*: Specifies if a 2-dimensional, or an almost 3-dimensional (2.5-dimensional), model of substrate roughness is used. See [subsection 4.11.2](#) for details. The 2.5-dimensional model is more realistic and the program default.
- *Calculation of dual scattering and roughness*: This switch has only influence if [Dual scattering](#) is selected (see [subsubsection 3.5.3](#)) and target or foil contain rough layers or a rough substrate (see [subsubsection 3.6.1](#)). If *Accurate* is selected, then an accurate simulation of dual scattering and roughness is performed. This simulation is usually very time consuming and in most cases unusable for practical applications. The selection of *Fast* reduces the computing time by about two orders of magnitude, but introduces a

3. Using SIMNRA

small inaccuracy in the dual scattering background. In most cases the *Fast* setting is fully sufficient and the program default.

The *File*-menu allows to save setups for calculations to disk and read setups from disk. All information contained in all tabs of the *Setup Calculation* form is saved. The *xnra* data file format is used for saving, the default file extension is *xcalc*.

- *Save as Default*: The current setup for calculation is stored as startup default for SIMNRA in the file *calc.xcalc*, see [Table 2.1](#). This affects all users of the computer.
- *Save Setup as*: Save the current setup to file.
- *Read Setup*: Reads a setup from file.

Note: You can read any *nra*-, *xnra*- or IDF-file with *Read Setup*. Only the calculation information will be read, any other information (such as target composition, experimental spectra etc.), which may be present in the file, will be ignored.

3.6. Target menu

3.6.1. Target

In this menu the target is created. A target consists of *layers*. Each layer consists of different elements and has a (mean) thickness. The composition of a layer does not change throughout its thickness. To simulate a concentration profile you have to use multiple layers. A layer may have roughness and may have porosity.

Layer number 1 is at the surface of the target, layer number 2 is below layer 1 and so on, see Figure 3.13.

- **Thickness:** Thickness of the layer (in 10^{15} atoms/cm²). For rough layers this is the mean layer thickness. The conversion factor from $\mu\text{g}/\text{cm}^2$ or μm to atoms/cm² can be determined with [Calculate: Density...](#) for pure elements, see [page 52](#).
- **Number of Elements:** Number of different elements in this layer. The maximum number of different elements in a layer is 40.
- **Element:** Name of the element, for example Si, W, Au. Lowercase and uppercase letters in element names are treated similar, so silicon can be entered as Si, si, SI or sI. XX means that this element is unknown. The special symbols D for deuterium, T for tritium and A for ⁴He can be used.
- **Concentration or areal density:** Atomic concentration or areal density of the element in the actual layer. The concentration c must be $0.0 \leq c \leq 1.0$. The sum of the concentrations of all elements in one layer must be equal to 1 ($0.999 \leq \sum c_i \leq 1.001$). If the sum of concentrations is not equal to 1, the word concentration is written in red colour, if the sum of concentrations is equal to 1, the word concentration is written in black colour. You can use the small buttons to set the concentration of the element i to 1 minus the sum of concentrations of all other elements, $c_i = 1 - \sum_{i \neq j} c_j$. The areal density is in 10^{15} atoms/cm². Whether concentration or areal density of an element is used, is determined by the [Concentration/Areal density](#) radio button, see [page 39](#).
- **Isotopes:** These buttons can be used to change the concentrations of isotopes of that element in the actual layer. This is only necessary if this element does not have natural composition of isotopes. You can create for example a layer of enriched ¹³C on top of ¹²C, or the like. The sum of concentrations of all isotopes of one element must be equal to 1.
Note: The [Isotopes](#) check-box in the [Setup: Calculation...](#) menu must be checked to manipulate individual isotopes.
- **Concentration/Areal density:** This radio button allows to select if amounts of individual elements are entered as atomic concentration or as areal density. If *Concentration* is selected, the total layer thickness and the concentrations of all elements are entered. If *Areal density* is selected, the areal density of each element is entered - the total layer thickness is calculated as the sum of all elemental areal densities.

3. Using SIMNRA

- *Correction factor(s) for stopping power:* SIMNRA uses Bragg's rule to calculate the stopping power of a layer, see [subsection 4.6.6](#) for more details. However, it has been shown experimentally that for several compounds like hydrocarbons or oxides deviations from Bragg's rule occur. To account for deviations from Bragg's rule a correction factor f can be used, and the program will use the stopping power $S(E)$ as function of energy E

$$S(E) = f S_{\text{Bragg}}(E) \quad (3.5)$$

$S_{\text{Bragg}}(E)$ is the stopping power according to Bragg's rule. Note that the factor f is energy independent. An individual factor f for each ion species and each layer may be defined. If no factor f is given the program uses $f = 1$, i.e. uses Bragg's rule.

- *Layer and substrate roughness:* Click the button if the current layer or the substrate is rough. See [subsubsection 3.6.1](#) for details.
- *Layer porosity:* Click the button if the current layer is porous. See [subsubsection 3.6.1](#) for details.

Manipulation of layers

To manipulate layers use the buttons in the *Layer manipulation* box. Additionally layers can be copied to and pasted from the clipboard and layers can be saved to and read from file.

- *Add:* Adds a layer. The added layer will be the last layer. The maximum number of different layers is 1000.
- *Ins:* Inserts a layer in front of the current layer. The maximum number of different layers is 1000.
- *Del:* Deletes the current layer.
- *Prev:* Go to the previous layer.
- *Next:* Go to the next layer.

Menu bar

- *File: Read Layer...:* Read a layer from file.

You can read any nra-, [xnra-](#), or IDF-file with *Read Layer...* Only the first target layer will be read, any other information (such as the composition of other layers, experimental spectra etc.), which may be present in the file, will be ignored.

Attention: If a layer is read from file, the current layer is overwritten.

- *File: Save Layer...:* Save the current layer description to file. The [xnra](#) data file format is used for saving, the default file extension is xlay.

- *File: Read Target...:* Read a whole target from file.

You can read any nra-, *xnra*-, or IDF-file with *Read Target...*. Only the target will be read, any other information (such as experimental setup, experimental spectra etc.), which may be present in the file, will be ignored.

Attention: If a target is read from file, the current target is overwritten.

- *File: Save Target...:* Save the current target to file. The *xnra* data file format is used for saving, the default file extension is *xtarget*.
- *File: Write Depth Profile...:* Save the depth profile of all elements in the target to file. The file format is shown in [Figure 3.14](#).
- *Edit: Copy Layer:* Copies the current layer to the clipboard.
- *Edit: Paste Layer:* Pastes a layer from the clipboard.

Attention: If a layer is pasted from the clipboard, the current layer is overwritten.

- *Edit: Copy Depth Profile:* Copies the depth profile of all elements in the target to the clipboard. The format is identical to a depth profile file, see [Figure 3.14](#). The depth profile can be pasted into an Origin or Excel worksheet.
- *Show: Target summary* displays the total amounts (in atoms/cm²) of all elements in the target. This is mainly useful if the target consists of plural layers composed of the same elements in different concentrations. The total amount of each element is the sum of this element in all layers.

Layer and substrate roughness

In this menu the roughnesses of the current layer and of the substrate are defined. See [section 4.11](#) for additional details.

- *Layer roughness distribution:* Defines the roughness of the current layer. If a layer is rough, then the layer thickness is not uniform, but varies from point to point. See [Figure 4.24](#) (left) for a schematic representation of a rough layer. A rough layer is described by a distribution function of layer thicknesses. This distribution is divided into N simulation steps. The step number N can be adjusted by the *Number of thickness steps* in the *Setup: Calculation...* menu, see [section 3.5.3](#).

The following distribution functions can be selected:

- *None:* The layer has no roughness, i.e. it is smooth and has no variation of layer thicknesses. This is the default for all layers.
- *Gamma:* A Gamma distribution of layer thicknesses is used, see [section 4.11](#) for details of this function. The Gamma distribution has 2 free parameters: mean value and standard deviation. The layer thickness is used as mean value. The standard deviation is defined by

3. Using SIMNRA

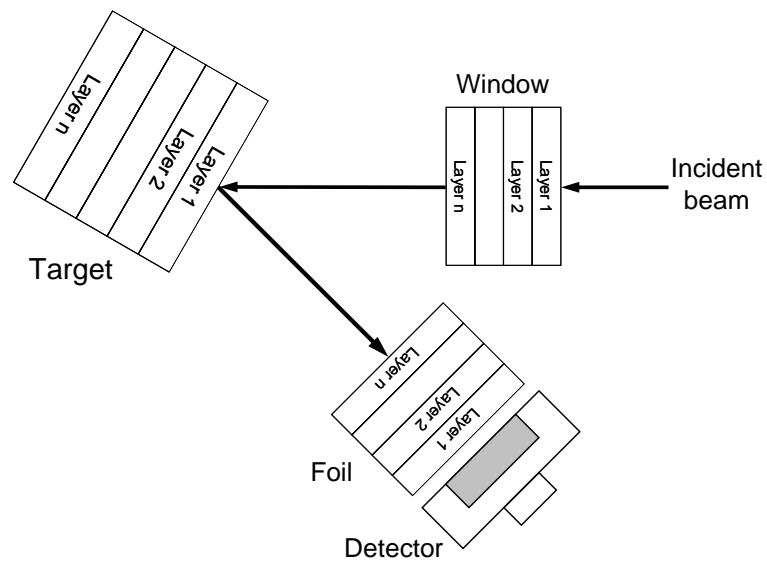


Figure 3.13.: Layer structure of target, foil and window. Target layer 1 is at the surface, the target layer with the highest number is the deepest layer. A window is penetrated by the incident beam before it hits the target. Window layer 1 is the first layer, and the window layer with the highest number is the last layer. Backscattered particles first penetrate the foil layer with the *highest* number, foil layer 1 is in front of the detector.

Depth	H	C
0	0.0	0.0
0	0.2	0.8
100	0.2	0.8
100	0.4	0.6
200	0.4	0.6
200	0.3	0.7
300	0.3	0.7
300	0.1	0.9
400	0.1	0.9

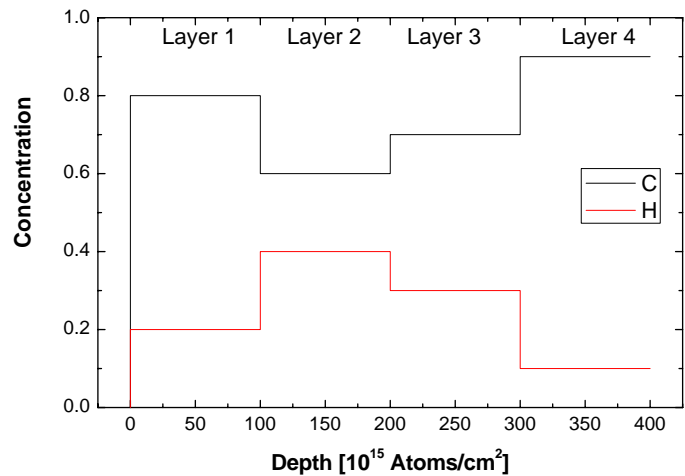


Figure 3.14.: Left: Example for a depth profile file. The target consists of 4 layers. The first line contains the names of the elements (H and C). The depth scale is in 10^{15} atoms/cm², the other columns contain the concentrations of the individual elements in the layers. The columns are separated by tabs. Right: Graphical representation of the depth profile.

```

This is a comment line
This is another comment line
0      0
999.9  0
1000   1
2000   1
2000.1 0
3000   0

```

Figure 3.15.: Example for a file defining a homogeneous frequency distribution of layer thicknesses from $1000\text{--}2000 \times 10^{15}$ atoms/cm².

FWHM of thickness distribution: The width and shape of the distribution is determined by the full width at half maximum (FWHM), with

$$\text{FWHM} = 2.35482 \sigma \quad (3.6)$$

where σ is the standard deviation ⁵. FWHM in 10^{15} atoms/cm².

- *From file:* The distribution function is taken from a user-supplied file. This is a very versatile method and allows to simulate almost any distribution of layer thicknesses, see [Figure 4.36](#) for some examples. The

File name (including full path) has to be supplied. Press the button in order to navigate to the file with the layer thickness distribution function. If the layer thickness distribution function is taken from a file, then the layer *Thickness* (see [section 3.6](#)) is taken as the mean value derived from the supplied function.

The file has to contain the frequency distribution of layer thicknesses, see [subsection 4.11.3](#) for more information. The file may contain an arbitrary number of comment lines at the beginning, followed by the frequency distribution as function of thickness. The thickness is given in 10^{15} atoms/cm². The frequency distribution will be normalized to unity by SIMNRA. An example for a valid frequency distribution file is shown in [Figure 3.15](#).

- *Substrate roughness distribution:* Defines the roughness of the substrate, on which the layers are deposited. See [Figure 4.24](#) (right) for a schematic representation of a layer on a rough substrate. A rough substrate is described by a distribution of local inclination angles φ (see [Figure 4.24](#) (right)), resulting in a distribution of incident and exit angles. Only one substrate can exist for all layers, i.e. the substrate parameters are identical for all layers, see [Figure 4.29](#). Substrate roughness is not available for foils. The distribution of incident and exit angles is divided into M steps. The step number M can be adjusted by the *Number of angular steps* in the *Setup: Calculation...* menu, see [subsection 3.5.3](#).

⁵Strictly speaking, [Equation 3.6](#) is only valid for a Gaussian distribution and not for the Gamma distribution. However, the Gamma distribution resembles a Gaussian distribution in many cases, which justifies to use the same FWHM. Internally, SIMNRA uses only the standard deviation σ , which is derived from the given FWHM through [Equation 3.6](#).

3. Using SIMNRA

The following distribution functions can be selected:

- *None*: The substrate has no roughness, i.e. it is smooth and has no variation of local angles. This is the default.
- *Gaussian*: The local angle distribution is described by a Gaussian function.
- *Lorentz*: The local angle distribution is described by a Lorentz function.
- *FWHM of substrate roughness*: Full width at half maximum (FWHM) of the local angle distribution, in deg.

Layer porosity

SIMNRA implements a model of porous layers assuming a random distribution of overlapping spherical pores in three dimensions. Pores are assumed to be hollow, i.e. to be empty without any material inside the pore. Porosity results in an additional energy spread of incident and outgoing particles.

- *Layer is porous*: Check if the current layer is porous.
- *Volume fraction of pores*: Volume fraction of pores in the current layer, in vol%.
- *Diameter of pores*: Diameter of the pores, in 10^{15} Atoms/cm² of the surrounding matrix material.
- *Variance per unit length*: This shows the variance of the path length distribution function inside pores per unit length. In 10^{15} Atoms/cm² of the surrounding matrix material.

3.6.2. Foil

In this menu a foil in front of the detector can be created. Like the target a foil can consist of multiple layers with different compositions. Foil layers can be rough and porous. If the foil consists of multiple layers, then backscattered particles first penetrate layer n , then layer $n - 1$ etc., layer 1 is directly in front of the detector (see [Figure 3.13](#)).

The default is no foil in front of the detector.

Some common materials used as stopper foils are already stored in the `\Layers` directory. The materials and files are listed in [Table 3.3](#). These files can be imported in the *Target* menu with *File: Read layer*.

3.6.3. Window

In this menu a window can be created. A window is traversed by the incident beam before it hits the target and is primarily intended to model the exit window for an external beam. Like the target and the foil a window can consist of multiple layers with different compositions, roughnesses and porosities. If the window consists of multiple layers, then the incident particles first penetrate layer 1, then layer 2 etc. until layer n , see [Figure 3.13](#).

The default is no window for the incident beam.

Table 3.3.: Predefined materials for stopper foils stored in the \LAYERS directory.

Filename	Material	Common name
INCON600.LAY	Inconel 600	
INCON625.LAY	Inconel 625	
SS14301.LAY	Stainless steel 1.4301, AISI 304 (US)	
SS14541.LAY	Stainless steel 1.4541	V2A
SS14571.LAY	Stainless steel 1.4571	V4A
SS316.LAY	Stainless steel 316 (US)	
MYLAR.LAY	Polyethylenterephthalat	Mylar, Hostaphan

Some common materials are already stored in the \Layers directory, see [Table 3.3](#). These files can be imported in the *Target* menu with *File: Read layer*.

3.7. Reactions menu

In the *Reactions* menu the cross-section data used for calculation of the simulated spectrum are chosen. Rutherford cross-sections for backscattering of projectiles and creation of recoils are available for all ion-target combinations, if kinematically possible. For heavy projectiles backscattered from light target nuclei two different solutions may be kinematically possible, see [Equation 4.4](#). The solution with the minus sign appears as *Rutherford cross-section (low energy solution)* in the *Reactions* menu.

The buttons *Rutherford* and *No Rutherford* allow a quick selection of Rutherford/Non-Rutherford cross sections for all isotopes of an element. The button *Rutherford* selects Rutherford cross sections for all isotopes and deselects all Non-Rutherford and nuclear reactions cross sections. The button *No Rutherford* deselects all Rutherford cross sections.

The buttons *SigmaCalc* and *No SigmaCalc* allow a quick selection of SigmaCalc cross sections for all isotopes of an element. The button *SigmaCalc* selects SigmaCalc cross sections for all isotopes of an element. It also selects Rutherford cross sections at low energies. Other cross-section data for the same reaction are deselected at overlapping energies, but remain selected at higher energies. Nuclear reactions cross sections are unaffected. The button *No SigmaCalc* deselects all SigmaCalc cross sections.

SIMNRA handles non-Rutherford cross-sections for scattering and recoil production and can use nuclear reactions cross-sections. SIMNRA reads cross-section data files in the [R33 file format](#). All cross-section data files in the Ion Beam Analysis Nuclear Data Library (IBANDL) (see <http://www-nds.iaea.org/ibandl/>) are stored in this format. See [subsection 3.17.1](#) and [Appendix D](#) for more details about this file format.

Add new cross-section data files is described in [section 3.17](#).

SIMNRA distinguishes between three different types of scattering events for each isotope:

1. Scattering of projectiles
2. Creation of recoils
3. Nuclear reactions.

The chosen cross-sections for each type of scattering event must be unambiguous: You can choose, for example, Rutherford cross-section for scattering in the energy range from 0.000–0.999 MeV, some non-Rutherford cross-section for backscattering in the energy range from 1.000–1.999 MeV and a different cross-section for backscattering in the energy range from 2.000–3.000 MeV. You cannot choose, however, Rutherford cross-section for backscattering in the range 0.000–2.000 MeV and another cross-section for backscattering in the energy range from 1.000–2.000 MeV: In this case the program does not know which cross-section it should use in the range from 1.000–2.000 MeV, and you will get the error message 'Energy overlap in cross-sections'.

Note 1: If a file contains scattering or recoil cross-section data as ratio-to-Rutherford, then SIMNRA assumes angular independence of the ratio-to-Rutherford values when using data initially obtained for a different angle. I.e. if a data file contains cross-section data as ratio-to-Rutherford values $\rho(E, \theta_0)$ for specific energies E and an angle θ_0 , then SIMNRA will use the

following cross-section values $d\sigma/d\Omega$ at the setup angle θ :

$$\frac{d\sigma}{d\Omega}(E, \theta) = \rho(E, \theta_0) \left(\frac{d\sigma}{d\Omega} \right)_R(E, \theta),$$

with $(d\sigma/d\Omega)_R$ the Rutherford cross-section at energy E and angle θ . In many cases this gives higher accuracy when using cross-section data valid for a slightly different angle.

If the cross-section data are given as direct values (i.e. in units of mb/sr), then SIMNRA will use these values as they are, i.e. without any further conversion.

Note 2: Some files contain total cross-section data $\sigma(E)$. In these cases the differentially cross-section $d\sigma/d\Omega$ is obtained by SIMNRA for all kinematically allowed angles by assuming angular independence of the cross-section in the center of mass system:

$$\left(\frac{d\sigma}{d\Omega} \right)_{CM}(E, \theta) = \frac{1}{4\pi} \sigma(E) \quad (3.7)$$

This cross-section is then transformed from the center of mass to the laboratory system. The assumption of angular independence in the center of mass system is well fulfilled for the $^3\text{He}(\text{D}, \text{p})\alpha$ -reaction for incident energies below about 1.2 MeV and for the $\text{D}(^3\text{He}, \text{p})\alpha$ -reaction for incident energies below about 1.8 MeV.

Note 2: Non-Rutherford cross-sections and nuclear reactions are only available if *Isotopes* in the *Setup:Calculation...* menu is checked.

3.7.1. Origin of cross-section data files

All cross-section data files in the "CrSec\IBANDL 2011" directory are retrieved from the Ion Beam Analysis Nuclear Data Library (IBANDL) which is maintained and provided freely by the Nuclear Data Services of the IAEA (<http://www-nds.iaea.org>). IBANDL is a product of an IAEA Research Coordination Project and while all efforts were made to ensure that the most accurate information was adopted, the IAEA accepts no responsibility for the usage of IBANDL data. References to the original publications are found in the file headers.

Cross-section data files from previous versions of SIMNRA are mostly contained in IBANDL, some additional files can be found in the CrSec\Additional directory. Cross-section data in the RTR (Ratio To Rutherford) file format are obsolete and no longer supported. These data files are now part of IBANDL.

3.7.2. Replacement of cross section data files

Throughout the long history of SIMNRA, cross section data files were renamed, reformatted (from RTR to R33), or replaced by more accurate data. To provide backward compatibility, the files REPLACE640.LST and REPLACE.LST are used. These files list the original cross section data file, which has been deleted or renamed, and the file it has been replaced with. If an NRA file is opened which references a renamed or replaced cross section data file, this file is automatically replaced and the new data are used for all calculations. A major change of cross-section data was introduced in version 6.40, when all previous cross-section data files were replaced by files

3. Using SIMNRA

from IBANDL. This replacement is monitored in the file REPLACE640.LST, while replacements in later versions are found in the file REPLACE.LST.

The format of the file REPLACE.LST is as follows:

OldFile, *NewFile*, *TagNr*

OldFile is the name of the renamed or replaced file and *NewFile* the name of the file it has been replaced with. *TagNr* is an integer number which is used only together with the file CRSDA.DAT - for any other file 0 should be used.

The file CRSDA.DAT, developed at the IPP Garching between 1985 and 1995, contained polynomial fit coefficients to various cross section data. It was deleted in SIMNRA 5.81: The accuracy of the fits was poor, and they were not documented. The fits were replaced by the original data files.

Cross-section data in the RTR (Ratio To Rutherford) file format became obsolete in SIMNRA 6.40 and are no longer supported. These files were replaced by corresponding files from IBANDL.

3.8. Calculate menu

In the *Calculate* menu all commands for calculating spectra, scattering kinematics, stopping powers and data fitting are located. Additionally some helpful tools (density conversions, particles*sr) can be found here.

3.8.1. Calculate Spectrum

Calculates the simulated spectrum.

3.8.2. Calculate Spectrum Fast

Calculates a simulated spectrum, but with decreased accuracy and faster integration algorithm. *Calculate Spectrum Fast* can be up to 2 times faster than *Calculate Spectrum*, but is generally less accurate.

3.8.3. Fit Spectrum...

Data fitting to experimental data.

In data fitting the quadratic deviation of the simulated from the measured data points

$$\chi^2 = \sum_i \frac{(N_{exp}(i) - N_{sim}(i))^2}{\sigma_i^2} \quad (3.8)$$

is minimized by varying the input parameters of the calculation. $N_{exp}(i)$ is the number of counts in the measured spectrum, $N_{sim}(i)$ the number of counts in the simulated spectrum, and σ_i the statistical error⁶. i can be either individual channels or channel regions, see [χ² evaluation](#).

⁶SIMNRA uses $\sigma_i = \sqrt{N_{exp}}$ for $N_{exp} > 4$, $\sigma_i = 2$ for $N_{exp} \leq 4$ due to Poisson statistics.

Fast fitting algorithms, such as the Levenberg-Marquardt algorithm, tend to be unstable and require the knowledge of the derivatives of χ^2 . SIMNRA uses the Simplex algorithm (sometimes called downhill simplex or Nelder-Mead method) for fitting [33, 87, 92]. The Simplex algorithm is very stable and converges (nearly) always. However, the convergence is not very fast. The Simplex algorithm always uses $n + 1$ points (called vertices) in the parameter space for fitting, where n is the number of free parameters.

You can fit:

1. Energy calibration (energy/channel and offset only, quadratic term is not changed)
2. Particles*sr
3. Composition and thickness of a layer
4. Roughness of a layer

independently or all at once. Check which parameters should be varied. Only one layer at a time can be fitted.

- *Number of fit regions*: Number of different regions where χ^2 is calculated. Up to 10 different regions may be used. At least one region must be specified. The regions should not overlap.
- *From To*: Lower and upper channel of each fit region.
- χ^2 *evaluation*: Determines how χ^2 (see Equation 3.8) is determined.

– *Channels*:

$$\chi^2 = \sum_i \frac{(N_{exp}(i) - N_{sim}(i))^2}{\sigma_i^2}, \quad (3.9)$$

where $N_{exp}(i)$ is the number of experimental counts in channel i , and $N_{sim}(i)$ the number of simulated counts in channel i .

– *Integrals*:

$$\chi^2 = \sum_i \frac{(N_{exp}(i) - N_{sim}(i))^2}{\sigma_i^2}, \quad (3.10)$$

where $N_{exp}(i)$ is the integrated number of experimental counts in region i , and $N_{sim}(i)$ the number of integrated simulated counts in region i .

If *Channels* is selected, then the shape of the spectrum is taken into account, resulting in more accurate results in most cases. If *Integrals* is selected, then only count integrals in specified regions are taken into account. This option should be selected, if the agreement between experimental and simulated spectra is poor due to inaccurate energy calibration, complicated peak shapes, and the like: In these cases, the selection of *Channels* may result in poor results, while *Integrals* still may work reliably.

The fit regions are identical to the regions of interest for χ^2 evaluation, see subsection 3.9.5.

3. Using SIMNRA

- *Max Iterations*: Maximum number of Simplex iterations. Fitting will be performed until the desired accuracy is obtained or the maximum number of iterations is reached.
- *Fit Accuracy*: Desired accuracy of the fit. The fit has converged if the relative change of all fitted parameters and of χ^2 is below *Fit Accuracy*. The relative change of a parameter A is $\Delta A/A$, where ΔA is the difference between the best vertex (the vertex with the lowest χ^2) and the worst vertex (the vertex with the highest χ^2).
- *Calculate Fit Error*: If checked an error estimate for all fitted parameters is computed. See [subsubsection 3.8.3](#) for details. This is a time consuming process and may require more computing time than the fit itself. Default is unchecked.

Fit error

An error bar for fitted parameters can be obtained under the following assumptions:

1. The physics model (i.e. stopping powers, cross sections etc.) is assumed to be accurate with zero error. Obviously, in reality this is not the case, and errors introduced by inaccurately known stopping powers or cross sections may largely exceed the computed errors. SIMNRA does not know whether a stopping power or cross section data set is accurate or not — this is left to the user.
2. All nonfitted parameters are assumed to be accurate with zero error. Again, usually this will be not the case, but SIMNRA is not able to quantify these errors: It does not know how accurately the energy calibration was determined or how accurately the ion current measurement was measured.

The error bar Δa of a fitted parameter a is determined by the shape of the χ^2 surface near the minimum: A flat minimum of χ^2 allows a larger variation of a and will result in larger errors of a . On a confidence level of 68.3% (the usual 1σ interval) the error of a is obtained by varying a until χ^2 has increased by 1 [92]:

$$\Delta\chi^2(a_{\Delta=1}) = \chi^2(a_{\Delta=1}) - \chi^2_{min} = 1. \quad (3.11)$$

χ^2_{min} is the minimised χ^2 with an optimised parameter a_{min} , and the error Δa of a is $\Delta a = |a_{\Delta=1} - a_{min}|$.

If we have more than one fit parameter things get more complicated [92]. As an example we consider the case of two fit parameters a_1 and a_2 . The $\Delta\chi^2 = 1$ contour now is an ellipse (Figure 3.16) due to correlations between a_1 and a_2 , and for more than two fit parameters a multi-dimensional ellipsoid. To find the confidence intervals for a_1 and a_2 we have to do the following: Increase a_1 by some amount (black arrows in Figure 3.16). Now find a new minimum of χ^2 by optimising a_2 , a_1 remains unchanged (dotted line in Figure 3.16). Increase a_1 and optimise a_2 again, and so on until $\Delta\chi^2 = 1$. For n fit parameters $a_1...a_n$ we have to increase a_1 and optimise $a_2...a_n$ until $\Delta\chi^2 = 1$.

As can be seen in Figure 3.16 the error bar $\Delta a'_1$ becomes nonrealistic small if only a_1 is fitted. This is a consequence of the assumption that all nonfitted parameters are accurate. To obtain realistic error bars all parameters should be fitted in one step.

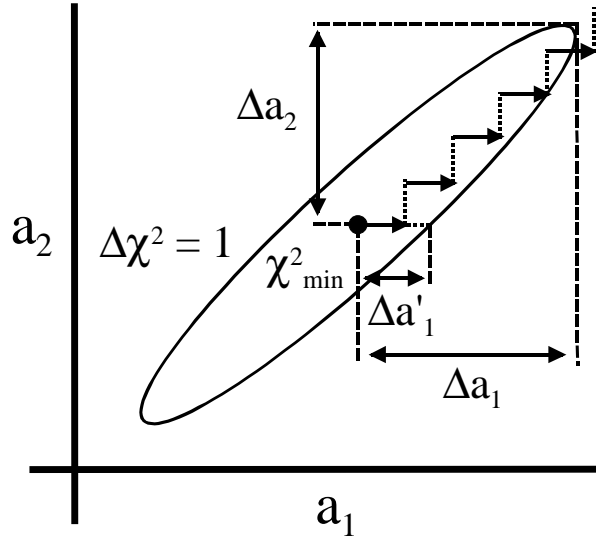


Figure 3.16.: $\Delta\chi^2 = 1$ contour near the χ^2 minimum (black dot) for two fit parameters a_1 and a_2 . Δa_1 and Δa_2 are the fit errors for a_1 and a_2 if fitted simultaneously, $\Delta a'_1$ the error for a_1 if only a_1 is fitted.

A quantitative measure for the goodness-of-fit of the assumed model can be obtained from the value of χ^2_{min} . See [92, Chapter 14.1] (or any good textbook about statistics) for more information.

3.8.4. Kinematics...

Calculation of scattering kinematics: Energies of scattered particles, recoils and nuclear reaction products.

3.8.5. Stopping...

Calculation of stopping powers for any projectile in any element and any target, foil or window layer. Calculation of energy loss and energy loss straggling in the different layers present in target, foil and window.

Energy losses in layers and target, foil and window are for normal penetration, i.e., perpendicular to the surface. Incident angle α and exit angle β , as defined in [Setup: Experiment](#) (see [subsection 3.5.1](#)), are ignored.

For a multi-layered target or window the projectiles are incident from the surface of layer 1 and then penetrate towards increasing layer numbers, see [Figure 3.13](#). But note that for a multi-layered foil the projectiles are incident from the rear side, i.e., they hit the foil at the rear side of layer n and then penetrate the foil toward decreasing layer numbers, see [Figure 3.13](#).

3.8.6. Cross Section...

Calculation of Rutherford cross sections for scattering and recoils.

3. Using SIMNRA

3.8.7. Density...

Density conversions (for elements only). Mass density to atomic density and conversion from atoms/cm² to μg/cm² and nm.

3.8.8. Particles*sr...

Calculation of particles*sr from the collected charge and detector solid angle. Often used solid angles can be saved for permanent use.

3.8.9. Exit Angle Beta...

Calculation of the exit angle β for IBM and Cornell geometries.

In IBM geometry incident beam, exit beam and the surface normal are in the same plane, see [Figure 3.17](#). The sample can be rotated in two different directions, so that for identical incident angle α and scattering angle θ two different values of β are possible. These two geometries can be called IBM 1 and IBM 2. In IBM 1 geometry the surface normal is rotated towards the direction of the exit beam, see [Figure 3.17](#), in IBM 2 geometry the surface normal is rotated away from the direction of the exit beam.

In IBM 1 geometry β is given by

$$\beta = |180^\circ - \alpha - \theta|.$$

In IBM 2 geometry β is given by

$$\beta = |180^\circ + \alpha - \theta|.$$

In Cornell geometry incident beam, exit beam and the sample rotation axis are in the same plane, see [Figure 3.17](#), and β is given by

$$\cos \beta = -\cos \theta \cos \alpha.$$

3.8.10. Subtract Pile-up...

Calculates the pile-up contribution not from simulated, but from experimental data, and allows to subtract the pile-up from the data. The pile-up is calculated iteratively from the *experimental* data (and not from *simulated* data, as is the case if *Calculate pile-up* in the *Setup: Experiment: More Options: Live Time and Pile-up...* form is checked, see [subsubsection 3.5.2](#)). This menu item can be used to subtract pile-up from any type of spectral data, including RBS or ERDA spectra, X-ray spectra (for example PIXE), or γ -ray spectra.

- The parameters *Real time*, *Live time*, *Pulse rise time*, *Fudge time parameter*, *Pile-up rejector*, and *Pile-up rejector pair-resolution time* are described in [subsubsection 3.5.2](#), see there for details.

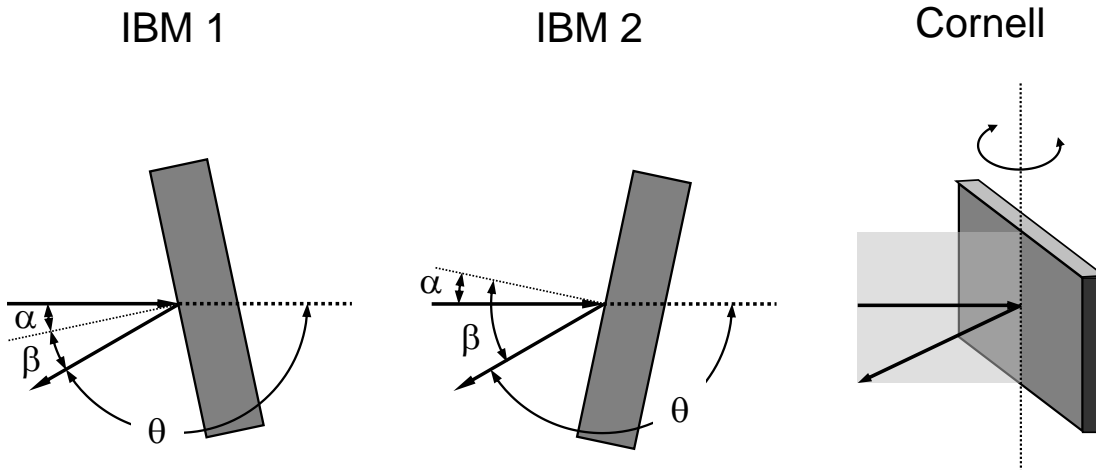


Figure 3.17.: IBM 1, IBM 2 and Cornell geometries with incident angle α , exit angle β and scattering angle θ . α and β are measured with respect to the surface normal indicated by a thin dotted line.

- *Number of iterations:* The pile-up contribution is calculated iteratively from the data. *Number of iterations* specifies the number of iterations. In most cases 2–3 iterations are sufficient.
- *Calculate pile-up:* Calculates the pile-up contribution from the experimental data.
- *Subtract pile-up:* Subtracts the pile-up contribution from the experimental data.
Warning: *Subtract pile-up* modifies irreversibly the experimental data.
- *Abort calculation:* Aborts a running calculation.

3.9. Tools menu

Several tools for spectrum evaluation are located in the *Tools* menu. Most tools appear as floating windows and are updated automatically if the spectrum is recalculated or a new spectrum is loaded from disk.

3.9.1. Data Reader

The data reader allows to read out the contents of a specific channel. The data reader is displayed as a small black crosshair. The data reader control window displays the channel, the mean energy of this channel, the lower and upper energy boundaries of the channel, and the number of counts in the channel. The data reader is moved by entering the channel number or by using the spin up/spin down buttons in the control window. The data reader is updated automatically if a spectrum is recalculated or if a new spectrum is loaded from disk.

3.9.2. Integrate Spectrum

The *Integrate Spectrum* tool allows to integrate a specific spectrum. The integral of a spectrum is the sum of all counts between a lower and an upper channel including the boundary channels. The integration boundaries are displayed as small black vertical lines in the plot. The boundaries are moved by entering the channel numbers or by using the spin up/spin down buttons in the control window. The integral is updated automatically if a spectrum is recalculated or if a new spectrum is loaded from disk.

3.9.3. Nearest Elements

The *Nearest Elements* tool marks the channels, in which scattered particles from specific elements or recoiled particles would appear. This allows easy identification of unknown peaks in the spectrum. The tool shows the three elements, which are closest to the cursor. The cursor is displayed as a black vertical line, and is moved by entering the channel number or by using the spin up/spin down buttons in the control window.

The *Nearest Elements* tool uses the surface approximation, i.e. stopping power effects are neglected. The marked channels are only valid for elements at the surface.

Limitations:

1. Stopping in a foil in front of the detector is neglected, resulting in unrealistic results if foils are present. A warning message is displayed, if a foil is present.
2. The tool assumes that particles are fully stopped in the detector, resulting in unrealistic results for thin detectors.
3. The tool displays energies of scattered particles and recoils, if kinematically possible. It does not display energies of nuclear reaction products.

3.9.4. Smoothed Experimental Spectrum

This tool draws a smoothed line through the experimental data points. The width of the smoothing window can be selected. See [section 3.21](#) for details of the smoothing algorithm.

3.9.5. χ^2 evaluation

χ^2 , i.e. the quadratic deviation between experimental and simulated data, is calculated in definable regions of interest. A click on χ^2 evaluation shows a form allowing to define the regions of interest. The χ^2 value is displayed in the statusbar, see [subsection 3.2.4](#). A double-click on the χ^2 value in the statusbar is identical to a click on χ^2 evaluation. The regions of interest for χ^2 evaluation are identical to the [fit regions](#), see [subsection 3.8.3](#).

- *Number of regions*: Number of different regions where χ^2 is calculated. Up to 10 different regions may be used. At least one region must be specified. The regions should not overlap.
- *From To*: Lower and upper channel of each region.
- χ^2 evaluation: Determines how χ^2 is determined.
 - *Channels*:

$$\chi^2 = \sum_i \frac{(N_{exp}(i) - N_{sim}(i))^2}{\sigma_i^2}, \quad (3.12)$$

where $N_{exp}(i)$ is the number of experimental counts in channel i , and $N_{sim}(i)$ the number of simulated counts in channel i .

- *Integrals*:

$$\chi^2 = \sum_i \frac{(N_{exp}(i) - N_{sim}(i))^2}{\sigma_i^2}, \quad (3.13)$$

where $N_{exp}(i)$ is the integrated number of experimental counts in region i , and $N_{sim}(i)$ the number of integrated simulated counts in region i .

If *Channels* is selected, then the shape of the spectrum is taken into account. If *Integrals* is selected, then only count integrals in specified regions are taken into account.

The reduced χ_r^2 is defined as

$$\chi_r^2 = \frac{\chi^2}{N} \quad (3.14)$$

with χ^2 according to [Equation 3.12](#) or [Equation 3.13](#) and N the number of channels (if *Channels* is selected) or the number of regions (if *Integrals* is selected). χ_r^2 is therefore χ^2 per channel (or per region).

A reduced χ_r^2 value close to 1 indicates an extraordinary good agreement between experimental data and simulation, and it implies that the difference between experimental and simulated data is only due to count statistics. This case is practically never achieved. Values of χ_r^2 between 2 and 5 usually indicate an acceptable agreement between experiment and simulation.

3. Using SIMNRA

3.9.6. *Resize Spectrum*

Allows to resize the experimental spectrum, i.e. to change the number of channels. The tool allows to resize the spectrum to any number of channels maintaining correct count statistics. In order to maintain correct count statistics an element of probability has to be introduced to the distribution of counts from the old to the new channels. Consequently the resized spectrum will show a different distribution of counts (within count statistics) each time the tool is used. This can be avoided by resizing by a power of 2 (i.e. decreasing the number of channels by a factor of 2, 4, etc.), for example by resizing a spectrum of 2048 channels to 1024 or 512 channels.

- *New number of channels*: Number of channels for the resized spectrum.
- *Adjust energy calibration*: If checked, then the *detector energy calibration* in *Setup: Experiment* is adjusted to the resized spectrum and the simulated spectrum is recalculated. If unchecked, then only the experimental spectrum is resized.

3.9.7. *Check Input*

Invokes an expert system for supporting the selection of the most accurate simulation settings for the given ion/target combination at the given energy and geometry. This expert system is a piece of artificial intelligence emulating the ability and knowledge of a human IBA expert. It examines the current settings of simulation parameters, points out potential problems, and recommends model settings with enhanced accuracy. See [section 4.14](#) for details.

3.10. Plot menu

This section describes all plot related commands. Most of these commands are also available through the context menu of the plot, which can be accessed by a right-click with the mouse on the plot.

- *Autoscaling*: If checked, the plot will be scaled automatically to minimum and maximum if experimental data are imported or a new calculation is performed. If unchecked, the axis scales remain fixed.
- *Rescale x-Axis*: Scales the x-axis to minimum and maximum of all visible spectra.
- *Rescale y-Axis*: Scales the y-axis to minimum and maximum of all visible spectra.
- *Unzoom*: Undo all zoom operations.
- *Show Spectrum*: Determines which spectra are shown in the plot.
 - *Show Experimental Spectrum*: If checked, the experimental spectrum is plotted. If unchecked, the experimental spectrum is not plotted. Default is checked.
 - *Show Smoothed Experimental Spectrum*: If checked, a smoothed experimental spectrum is plotted. See [section 3.21](#) for details about the smoothing algorithm. If unchecked, the smoothed spectrum is not plotted. Default is unchecked.
 - *Show Simulated Spectrum*: If checked, the simulated spectrum is plotted. If unchecked, the simulated spectrum is not plotted. Default is checked.
 - *Show Simulated Element Spectra*: If checked, individual simulated spectra for each element in the target are plotted. The spectrum is only plotted if it contains counts, i.e. spectra with zero counts are not shown. Default is unchecked.
 - *Show Simulated Isotope Spectra*: If checked, individual simulated spectra for each isotope in the target are plotted. The spectrum is only plotted if it contains counts, i.e. spectra with zero counts are not shown. Default is unchecked.
- *x-Axis...*: Scale the x-axis manually by entering the axis minimum and maximum, set logarithmic x-axis. Same as a double-click with the left mouse button on the x-axis.
- *y-Axis...*: Scale the y-axis manually by entering the axis minimum and maximum, set logarithmic y-axis. Same as a double-click with the left mouse button on the y-axis.
- *Spectrum...*: Allows to change the appearance of individual spectra by setting line colour and symbol shape. Same as a double-click with the left mouse button on any spectrum.
- *Legend...*: Allows to alter the text of the legend. Same as a double-click with the left mouse button on the legend.
 - *Legend outside of chart*: If checked, then the legend is placed to the right outside of the chart. This is useful to avoid overlap of the legend with spectra. If unchecked, then the legend is placed inside the chart. This gives more space to spectra for easier viewing.

3. Using SIMNRA

- *Delete Experimental Data*: Deletes the experimental data from the plot.
- *Delete Simulated Data*: Deletes all simulated data from the plot.
- **Zooming into the plot**: To zoom into the plot click with the left mouse button into the upper left corner of the range you want to zoom in. Keep the mouse button down and tear a rectangle to the lower right corner of the zooming range.
- **Zooming out**: Click with the left mouse button into the plot. Keep the mouse button down and move the mouse towards the upper left corner. Or use *Plot: Unzoom*.
- **Panning**: Press the *Shift* key and click simultaneously with the left mouse button into the plot, keep the mouse button down and move the mouse.
- **Context menu**: Click with the right button into the plot. The commands available in the context menu are identical to the *Plot menu* commands.
- **Context menus of plot components**: Context menus of the x-axis, y-axis, all spectra, and the legend are accessed by a double-click on the component. The corresponding context menus are identical to the menu items accessed through the *Plot* menu or through the *context menu* of the plot.

3.11. Options menu

3.11.1. Create List of Cross-Sections

SIMNRA uses a file named CRSEC.LST in the cross-sections directory to know which cross-section data are available. *Create List of Cross-Sections* will create this file. You have to recreate the reaction list if you add or delete cross-section data files, see [section 3.17](#).

In some cases files are not readable by SIMNRA due to format errors. These files will be ignored. The program displays a list of all ignored files.

3.11.2. Preferences...

Allows to set global program preferences. These preferences are stored permanently.

Print tab

- *Show "Print What?" dialog*: If checked, then a dialog asking what to print (experimental conditions, graph) is displayed. If unchecked, this dialog is omitted and everything is printed.

Spectrum Data tab

- Determines which menu entries are visible in the *File: Read Spectrum Data* menu, see [subsection 3.3.5](#).

Saving tab

- *Create backup file when saving*: If checked, then the old xnra-file is saved to a file named backup.xnra each time *File: Save* or *File: Save As...* is used. In case of erroneous overwriting of a file the old data can be recovered from this file. In each directory only one backup.xnra can exist.

Directories tab

- Determines in which directories the program looks for atomic, cross-section and stopping power data. This is mainly useful if the data are stored on a network drive and are accessed simultaneously from different computers. Changing these directories requires running SIMNRA with administrator privileges by right-clicking the program and then selecting *Run as administrator*.
- *SRIM program directory*: Base directory of the SRIM program for stopping power calculations. See [section 3.18](#) for details. This can be changed for *All users* by running SIMNRA as administrator (by right-clicking the program and then selecting *Run as administrator*) or for the *Current user* by running SIMNRA without administrator privileges.

3. Using SIMNRA

3.12. Help menu

3.12.1. User's Guide

Opens the User's Guide in PDF-format using the default PDF-viewer.

The use of Adobe Acrobat or Adobe Reader/Adobe Acrobat Reader DC is recommended. These are not part of SIMNRA. See [section 2.1](#) for details. The SIMNRA help system has limited functionality with alternative PDF-viewers.

3.12.2. Home Page

Opens the SIMNRA home page using your default web browser.

3.12.3. IBANDL Cross-Sections

Opens the Ion Beam Analysis Nuclear Data Library (IBANDL) home page using your default web browser. IBANDL contains a large number of cross-section data files for ion beam analysis and is hosted by the International Atomic Energy Agency (IAEA).

3.12.4. Mott Cross-Sections

Runs the program *MottCalc* [48] for calculating Mott scattering cross-sections for identical particles in forward direction. The scattering of identical particles (for example ^{12}C on ^{12}C) results in a quantum-mechanical interference term and the scattering- and recoil cross-sections deviate from Rutherford already at low energies, see [subsubsection 4.3.3](#) for details.

MottCalc calculates the Mott scattering cross-sections for a number of projectile-target combinations. In order to use Mott scattering cross-sections perform the following steps:

1. Click [Help: Mott cross-sections](#) for running *MottCalc*.
2. Select mass A and nuclear charge Z of the isotope.
3. In *Mode Selection* select *Energy Distribution*. Select the desired *Detection Angle* and energy range.
4. Click the button *Save as R33 file*. Store the file in the directory \CrSec\User of your SIMNRA installation.
5. Recreate the reaction list in SIMNRA by clicking [Options: Create List of Cross-Sections](#). See [section 3.17](#) for further details and trouble shooting.

MottCalc has been developed at the National Technical University of Athens, Athens, Greece [48] and was provided by M. Kokkoris.

3.12.5. About...

Shows the version number of the program and some additional information.

3.12.6. Register...

Shows the registration status of SIMNRA on the current computer and allows to register it.

The registration status of SIMNRA can be *Unregistered*, *Registered for current user*, or *Registered for all users*.

- *Unregistered*: SIMNRA is unregistered if no or an invalid registration number was entered.
- *Registered for current user*: SIMNRA is registered only for the current user of the computer. For other users it will be *Unregistered*.
- *Registered for all users*: SIMNRA is registered for all users of the computer.

Registering SIMNRA requires entering a valid registration number into the *Registration!registration number* field. See [home.mpcdf.mpg.de/ mam/Info.html](http://home.mpcdf.mpg.de/mam/Info.html) for contact information to the author in order to obtain a valid registration number. The registration number for *All users* is only visible if SIMNRA is run with administrator privileges, see below.

SIMNRA can be registered for

- *Current user*: SIMNRA is registered only for the current user of the computer. Registering for *Current user* does not require special privileges.
- *All users*: SIMNRA is registered for all users of the computer. This item is only available if SIMNRA is run with administrator privileges by right-clicking the program and then selecting *Run as administrator*. This requires administrator privileges on the computer.

3.13. PIGE

SIMNRA can run in two different modes:

- *NRA mode*: In this mode energy spectra of charged particles (scattered particles, recoils, or nuclear reaction products) are calculated [73, 75]. *NRA mode* is the default.
- *PIGE mode*: In this mode particle induced gamma ray emission (PIGE) is calculated by simulating total yields of gamma lines [13].

SIMNRA can be set to *PIGE* mode by using one of the following methods:

- Running SIMNRA with the command line parameter */PIGE* or *-PIGE*. This is installed as *Simnra for PIGE* in the *PIGE* sub-directory of the *Simnra* start menu item.
- Opening an xnra-file with PIGE data.
- Using *OLE automation* by setting *App.SimulationMode* = 1.

PIGE mode behaves differently from *NRA* mode:

- Only PIGE cross-sections are available, cross-sections for charged particles are not displayed and cannot be selected.

3. Using SIMNRA

- Energy spectra are not calculated or displayed, only total yields of different gamma ray lines.
- Experimental gamma-ray spectra or gamma-ray yield data cannot be read.
- Fitting is not possible.

3.14. Neutron-induced reactions and neutron-ERD

SIMNRA version 7.03 and higher can calculate energy spectra of charged particles created by neutron-induced nuclear reactions (for example by the ${}^6\text{Li}(n,\alpha)t$ reaction) or by neutron-ERD (sometimes called n-ERD or NERDA, i.e. charged particles recoiled by high-energetic incident neutrons). n-ERD has been used to determine depth profiles of hydrogen isotopes until large depths [37, 63, 103].

Incident neutrons can be selected by clicking the corresponding radio button in the *Setup:Experiment* form, see [subsection 3.5.1](#).

Neutron transport in matter is treated in a simplified way:

- Neutron trajectories in matter are approximated by straight lines.
- Energy loss and energy straggling of neutrons in matter is zero, i.e. neutrons simply keep constant energy along their trajectory.
- Unless otherwise specified incident neutrons are assumed to be mono-energetic. A Gaussian (or almost Gaussian) incident energy spread can be used, see *Energy spread of incident beam* in [subsection 3.5.1](#).
- Angular spread of incident neutrons is neglected, i.e. a parallel incident neutron beam is assumed.

For calculating a charged-particle spectrum emitted by neutrons cross-section data for nuclear reactions or recoil creation are necessary. Currently SIMNRA provides only a very limited set of cross-section data for neutrons. See [section 3.17](#) how to add additional cross-section data.

3.15. Data exchange with other programs

3.15.1. Graphics programs: Excel, Origin, ...

SIMNRA allows to exchange data with graphics programs by two different methods:

1. Via the clipboard. With *Edit: Copy Data* experimental and simulated spectra are copied in ASCII format to the windows clipboard. From there these can be pasted into any spreadsheet program. See [subsection 3.4.1](#) for details.
2. Via ASCII file. With *File: Write Spectrum Data...* experimental and simulated data are exported as columns to an ASCII file. This file can be imported into any plot program, such as Excel, Origin or Mathematica. See [subsection 3.3.6](#) for details.

3.15.2. RUMP

SIMNRA can read and write sample description files (*.LCM) and read RBS files (*.RBS) in binary and ASCII format used by the RUMP program.

Sample description files contain the composition of the sample and the absorber foil. These files can be read and written by *File: RUMP: Read Sample Description File* (see [section 3.3.7](#)) and *File: RUMP: Write Sample Description File* (see [section 3.3.7](#)).

RUMP stores experimental parameters (Type of incident particles, incident energy, scattering geometry, etc.) and spectral data in files with extension *.RBS. These files can be read by *File: RUMP: Read RBS File*, see [section 3.3.7](#). SIMNRA determines automatically if the file is binary or ASCII. SIMNRA can not write RBS files.

The parameters imported by SIMNRA from RUMP RBS files are listed in [Table 3.4](#).

SIMNRA and RUMP use a different naming convention for the three angles (incident angle, scattering angle, exit angle) which define the geometry of an experiment:

- The incident angle is called α in SIMNRA, and Θ in RUMP.
- The scattering angle is called θ in SIMNRA, while RUMP uses the supplement of the scattering angle $\phi = 180^\circ - \theta$: Direct backscattering is $\theta = 180^\circ$ in SIMNRA, but $\phi = 0^\circ$ in RUMP.
- The exit angle is called β in SIMNRA, and ψ in RUMP. This angle is always independent in SIMNRA (i.e. SIMNRA always uses general geometry), while RUMP determines this angle from $\psi = \phi + \theta$ in IBM geometry or $\cos(\psi) = \cos(\phi) \cos(\theta)$ in Cornell geometry.

These relations are summarized in [Table 3.5](#). See [Figure 3.6](#) for a graphical representation of the different angles. SIMNRA adjusts the angles correctly if a RUMP RBS-file is opened.

Sample Description Files

SIMNRA supports only a subset of the RUMP sample description commands. The supported command are listed in [Table 3.6](#). All other commands will be neglected. Note that especially the RUMP commands *Equation*, *Species* and *Fuzz* are not supported. If your sample description file contains these commands, they will be neglected and a warning will be shown.

3. Using SIMNRA

Table 3.4.: Parameters imported by SIMNRA from RUMP binary and ASCII RBS files. A — means that this parameter is not supported. A • in the *RUMP Binary* column means that this parameter is fully supported. The *RUMP ASCII* column lists the supported commands in ASCII files. These commands can be abbreviated to the capitalized part. See www.genplot.com/doc/RUMP/rbs_inf.htm for the RUMP Binary Data Format Specification and www.genplot.com/doc/RUMP/cmds.htm for the RUMP command reference.

	RUMP Binary	RUMP ASCII
Real time/Live time ^a	•	—
Date and time of data collection	•	—
Incident beam energy	•	MEV
Incident ion species ^b	•	BEAM
Charge state of incident beam ^c	•	BEAM
Integrated beam charge	•	CHARGE
Incident beam current	•	CURRent
Conversion factors channel to energy	•	CONVersion
FWHM of detector resolution	•	FWHM
Geometry	•	GEOMetry
Incident angle Θ	•	THEta
Exit angle ϕ	•	PHI
Scattering angle ψ	•	PSI
Starting channel of data	•	CHOFF
Correction factor	•	CORrection
Spectrum data	•	SWALLOW

^a*RUMP ASCII* files do not offer a realistic possibility to store real time and live time. For these files SIMNRA estimates the real time from the integrated beam charge and the incident beam current. *RUMP Binary* files can store real and live times as string in the form 'LT=80.2 CT=90.0', with *LT* the live time and *CT* (clock time) the real time. Both values in seconds. See the *RUMP Binary Data Format Specification* for *Record type 0102h*.

^bThe ion mass should be indicated by preceding the ion name, for example 4He++ or 1H+. The most abundant isotope is assumed if the mass is omitted.

^cIn ASCII files the charge state of the incident beam is indicated by the number of + signs in the BEAM command, for example BEAM 4He++ for a double charged beam. A charge state of 1 is assumed if the charge state is omitted.

Table 3.5.: Naming conventions for the geometry of an experiment, as used by SIMNRA and RUMP, and the relations between them.

	SIMNRA	RUMP	Relation
Incident angle	α	Θ	$\alpha = \Theta$
Scattering angle	θ	ϕ	$\theta = 180^\circ - \phi$
Exit angle	β	ψ	$\beta = \phi + \Theta$ (IBM geometry) $\cos \beta = \cos \phi \cos \Theta$ (Cornell geometry) $\beta = \psi$ (General geometry)

Table 3.6.: RUMP sample description commands which are supported by SIMNRA.

RESET
LAYER
OPEN
NEXT
COMPOSITION ^{ab}
THICKNESS ^c
ABSORBER

^aPartly implemented. Only elements are recognised, but no isotopes. The command *Composition Si 1 O 2 /* will be recognised and the natural isotopic ratios for Si and O will be used. The command *Composition 28Si 1 16O 2 /* will not be recognised properly.

^bRUMP allows to enter SiO₂ as Si 1 O 2. SIMNRA will convert this to Si 0.33333 O 0.66666.

^cAs units for thickness are possible: /CM2, M/CM2, A, NM. If A for Å or NM for nm are used, SIMNRA will use the weighted atomic densities of all elements in the layer to convert Å or nm to 10¹⁵ atoms/cm². It is highly recommended to use the units /CM2 or M/CM2 instead.

RBS Files

SIMNRA can read RUMP's RBS file format version 1.1 from 8/94 with the following limitations:

1. Data compression level 3 (zero compressed) is not implemented. Levels 0–2 (uncompressed real, uncompressed integer, differential integer) are fully implemented.
2. Each RBS file may contain only one spectrum. Record type 20h is allowed, but may contain only one row.
3. Only one record 120h (RBS spectrum type) or 121h (FRES spectrum type) may be present, i.e. simultaneous RBS and FRES (= ERD) are not allowed.
4. Record types 01h (printed comments), 02h (unprinted comments), 101h (identifier for the data set), 102h (MCA information) and 103h (collection date/time) are recognized, but ignored.
5. Record type 111h (accelerator parameters): Beam current is ignored by SIMNRA, and pile-up is not calculated.
6. Record type 112h (data collection parameters): Starting channel of data is rounded from real to integer. This may result in differences of ± 0.5 channels between RUMP and SIMNRA, if a non-integer starting channel is used.

3.15.3. IBA data furnace

The IBA data furnace (NDF, WiNDF) was developed at the University of Surrey and is available from www.ee.surrey.ac.uk/Research/SCRIBA/ndf/. SIMNRA and NDF use both the IBA data format IDF [17] for storing ion beam analysis data. This allows easy exchange of all data between both programs.

3.16. Importing spectrum data in any format

SIMNRA can read experimental spectrum data in several formats including a number of ASCII formats, see [subsection 3.3.5](#). But many laboratories have their own spectrum data file formats. SIMNRA offers the possibility to import any type of experimental data by supplying a dynamic link library (dll), which reads the data and passes them to SIMNRA. This dll must be supplied by the user and is called if [File: Read Spectrum Data: User...](#) is clicked. This section describes the details of this dll.

The dll-name must be `user.dll`. It must be located in the `UserDll` subdirectory - otherwise SIMNRA will not find it. The dll must be 64-bit for a 64-bit version of SIMNRA and 32-bit for a 32-bit version of SIMNRA.

The dll must export a function *ReadData* (exact spelling), defined as follows⁷:

```
Function ReadData (FileName : PChar;  
                  Var Count : Integer;  
                  Data : Pointer): Integer; stdcall;
```

Filename Input parameter. Null terminated Unicode string with the full name including path to the file which shall be read. PChar values are word-sized (16 bits) Unicode characters.

Count Output parameter. Pointer to a 32 bit signed integer value which indicates how many channels were actually read.

Data Input parameter. Pointer to an array of 32 bit signed integer values which will take the spectrum data. The array starts at channel 0(!) and has a maximum of 8192 channels.

The return value of *ReadData* is a 32 bit signed integer. It must be 0, if the file was read successfully. Any other return value indicates an error condition.

The calling convention of *ReadData* must be *stdcall* in Borland Delphi or *WINAPI* in Microsoft C++.

A small code example in Pascal can be found in `UserDLL\sample.dpr`. A more detailed example is given in `UserDLL\sample1.dpr`.

⁷The code examples are in Pascal.

3.17. Adding new cross-section data

To add new cross-section data, you have to perform the following steps:

1. Create a cross-section data file in the R33 file format. The easiest way to do this is by using the R33Manager. Alternatively you can use any text editor. The file format is described below.
2. Copy this file into the directory \CrSec\User of your SIMNRA installation.

Note: File names in the \CrSec directory and all of its sub-directories must be unique, i.e. all files in the \CrSec\User directory must have different names than files in the "\CrSec\IBANDL 2011" and \CrSec\Additional directories.

3. Recreate the reaction list by clicking *Options: Create List of Cross-Sections*.

Note: If your file is ignored, SIMNRA was not able to read or understand the file. Carefully read [subsection 3.17.1](#) about the [R33 file format](#) and try again.

The nuclear data section of the International Atomic Energy Agency (IAEA) has created the **Ion Beam Analysis Nuclear Data Library IBANDL** at <http://www-nds.iaea.org/ibandl/>. This library contains many cross-section data files and is constantly updated. The data files are in R33-format.

3.17.1. The R33 file format

The R33 file format is described in full detail by I.C. Vickridge in [Appendix D](#). SIMNRA reads the updated R33 file format of April 2002, but is also able to read older R33 files which conform to the original specification from the year 1991⁸.

An example for a valid file in the R33 format is shown in [Figure 3.18](#). Each line must end with <CR><LF> (Carriage Return and Line Feed).

SIMNRA uses not only the data points, but also a part of the information supplied in the file header. The following lines are used by SIMNRA and must be present in the file. Though the lines may be arranged in any order, it is recommended to arrange them in the same order as shown in [Figure 3.18](#). Any other entries than the ones listed below are ignored by SIMNRA, but may be necessary to form a valid R33 file. See [Appendix D](#) for details.

- A line containing the string **Source:**⁹. The rest of the line has to contain a reference or other source for the data and will appear as description in the reaction menu.
- A line containing the string **Reaction:**. SIMNRA will interpret the nuclear reaction string written in that line (In the example of [Figure 3.18](#) $^{160}\text{d}, \alpha^{14}\text{N}$) to find out which particles are involved in the nuclear reaction. The masses of the particles are ignored.

⁸The original specification can be found in DSIR Physical Sciences Report 33 by I.C. Vickridge

⁹Note that in this and all following strings the concluding colon : belongs to the string and must not be omitted.

3. Using SIMNRA

COMMENT: These cross sections have been digitised from the publication cited below. No error of either the energy and or the sigma is given. Some errors may be among the data, we are recently checking them. The Los Alamos Ion Beam Handbook also will contain these data as soon as it is ready. If you use this data please refer to the paper below. Source: A.Turos, L.Wielunski and a Batcz, NIM, 111(1973), 605
Special comment:
WARNING ! THIS IS MAINLY FOR TEST NO GUARANTY IS PROVIDED FOR EVEN AGREEMENT WITH THE ORIGINAL PUBLICATION.
File created by R33 Manager version 0.1

Version: R33
Source: A.Turos, L.Wielunski and a Batcz, NIM, 111(1973), 605
Name: Gyorgy Vizkelethy
Address1: Department of Physics
Address2: Idaho State University
Address3: Campus Box 8106
Address4: Pocatello, ID 83209-8106
Address5: (208) 236-2626
Address6: vizkel@physics.isu.edu
Serial Number: 0
Reaction: $160(d,a0)14N$
Distribution: Energy
Composition:
Masses: 2.000, 16.000, 4.000, 14.000
Zeds: 1, 8, 2, 7
Qvalue: 3110.00, 0.00, 0.00, 0.00, 0.00
Theta: 145.00
Sigfactors: 1.00, 0.00
Enfactors: 1.00, 0.00, 0.00, 0.00
Units: mb
Data:

761.0	0.0	2.92E+0000	0.0
770.0	0.0	3.65E+0000	0.0
775.0	0.0	3.99E+0000	0.0
780.0	0.0	4.41E+0000	0.0
785.0	0.0	4.55E+0000	0.0

EndData:

Figure 3.18.: Example for a cross-section data file in the R33 file format.

- A line containing the string `Masses:`. SIMNRA will read the masses of the particles from this line. All masses in amu. The first mass is the mass of the incident particle, the second mass is the mass of the target particle, the third mass is the mass of the outgoing particle for which the cross-section is valid and the fourth mass is the mass of the other reaction product. The masses may be rounded to the nearest integer value - SIMNRA will replace the given masses by the exact values.

Warning: Note that the masses are given in a different order than in the `Reaction:` string! For the reaction `160(d,a0)14N` the correct line is `Masses: 2, 16, 4, 14`. This is a permanent source of error, so check your input carefully.

In some cases, for example if the file contains elastic scattering data, the cross section values are not for a specific isotope, but for natural isotopic composition. In this case the mean target mass should be used.

- If the cross section values are for natural isotopic target composition rather than for a specific isotope a line containing the string `Composition: Natural` has to be present. In this case the given cross section is used for all isotopes. This line has to be omitted if the cross section is for a specific isotope.

Other compositions than `Natural` must not be used.

- A line containing the string `QValue:`. The Q-value is the energy released in the nuclear reaction (in keV). $Q = 0.0$ for elastic scattering. Up to 5 different Q-values are allowed, for example for multiple particle groups which are not resolved. SIMNRA will use the mean value of all non-zero Q-values.
- A line containing the string `Theta:`. Theta should be given in degrees. The value of theta is not used by SIMNRA, but this line must be present and contain a value.
- Optionally a line containing the string `Units:` may be present. Valid values are `mb` for differential cross sections, `tot` for total cross sections, and `rr` for ratio to Rutherford. Differential cross sections have to be in mbarn/sr, and total cross sections in mbarn. If the `Units` line is omitted, SIMNRA will assume `mb`, i.e. differential cross sections. The value `tot` indicates that the cross section is integrated over all angles. Thus if `tot` is used the value of theta has no meaning. Nevertheless, a valid real number has to be given for theta.
- A line containing either the string `Nvalues:` or `Data:`. The value of `Nvalues:` is ignored. SIMNRA assumes that the data will start after this line.
- The data are organised in 4 columns: The first column is the energy in keV, the second column is the energy error in keV (ignored by SIMNRA), the third column is the cross-section in the laboratory frame, and the fourth column is the cross-section error (ignored by SIMNRA). The cross section units have to be mbarn/sr for differential cross sections and mbarn for total cross sections. SIMNRA expects the data to be arranged in order of ascending energy.

3. *Using SIMNRA*

- Scale conversion factors (EnFactors and SigFactors) may be used with SIMNRA 5.02 and higher. Refer to [Appendix D](#) how to use scale conversion factors.

3.18. Using SRIM stopping powers

SRIM is a program for the calculation of stopping powers and ion ranges. SRIM is not part of SIMNRA, but is developed by J. Ziegler and can be downloaded from www.srim.org. SIMNRA uses SRIM for calculating stopping power tables, which are then used for simulations. SIMNRA requires SRIM 2003 or higher.

In order to use SRIM stopping powers, you must perform the following steps:

1. Download and install SRIM on your computer.
2. The directory [SRIM]\SR Module (with [SRIM] the directory where SRIM is installed) must provide read/write access. This is usually not the case if SRIM is installed in the \Programs or \Programs (x86) branch. Set read/write access to [SRIM]\SR Module manually or install SRIM in a different directory, for example C:\SRIM.
3. Run SIMNRA. Click *Options: Preferences...* and go to the *Directories* tab. Enter the path to the *SRIM program directory* (i.e. the directory where SRIM 20nn.exe is located. nn is the year, such as 2003, 2008, etc.). You can use the small button to navigate graphically to that directory.
4. Click *Setup: Calculation...* and go to the *Parameter* tab. Select SRIM as *Electronic stopping power data*.

SIMNRA uses SRIM to calculate stopping power tables. These tables are stored in the Stop directory. The file naming convention is SRIM2003_Z1_M1_Z2.dat, where Z1 and M1 are the nuclear charge and mass of the projectile, and Z2 the nuclear charge of the target material. Initially there are no SRIM stopping power files available, but these files will be created successively whenever you are running a simulation.

3.18.1. Trouble shooting

If SIMNRA is not able to use SRIM stopping powers, please read the error message carefully and act according to the following list.

- Error message: *Error running SRIM module*.
 1. Is SRIM 2003 or higher installed on your computer? See [section 3.18](#) on how to install SRIM.
 2. Check the path to the SRIM program directory in the *Options: Preferences: Directories* form. See [section 3.18](#) for details.
 3. SIMNRA calls the program [SRIM]\SR Module\SRModule.exe, where [SRIM] is the SRIM program directory from the *Options: Preferences: Directories* form. This program must exist, and it must be located in the \SR Module subdirectory (note the blank between SR and Module). Re-install SRIM, if this program does not exist.

3. Using SIMNRA

- Error message: *Error creating SRIM Module input file. Cannot write to file xyz.*

SIMNRA must have write access to the [SRIM]\SR Module subdirectory, and it must have write access to the file [SRIM]\SR Module\SR.IN, where [SRIM] is the SRIM program directory from the [Options: Preferences: Directories](#) form. This may be a problem if SRIM is installed in the \Programs branch. This is a bug of SRIM, which is not foreseen to be used with restricted write permissions – complain to the SRIM author. You must grant write access to the [SRIM]\SR Module subdirectory. The easiest solution is the installation of SRIM outside of the \Programs branch, for example in C:\SRIM.

- Error message: *Error creating SRIM stopping power data file. Cannot write to file xyz.*

SIMNRA must have write access to the [SIMNRA]\Stop directory, where [SIMNRA]\Stop is the stopping power data directory in [Options: Preferences: Directories](#). The SIMNRA setup program grants write access to the [SIMNRA]\Stop directory to all registered users, but somebody may have changed that later. You must grant write access to the [SIMNRA]\Stop directory.

Note: Unrestricted write access to the [SIMNRA]\Stop directory does not cause a security problem, because this directory contains only data files.

- Anything else: Check the files SRIM2003_Z1_M1_Z2.dat in the SIMNRA\Stop directory. Delete all files with a length of 0 byte.

```

This is a comment line
This is another comment line
E      Stopping
(keV)  (keV/1E15 at./cm2)
1      0.5
2      0.6
4      0.7

```

Figure 3.19.: Example of a valid stopping power data file.

3.19. Using user-defined stopping powers

SIMNRA will use user defined stopping powers, if *User defined* stopping powers are selected in *Setup: Calculation*, see subsection 3.5.3.

To use user defined stopping powers, you have to perform the following steps:

1. Stopping power data files must be stored in the [SIMNRA]\Stop directory.
2. Stopping power data files are selected according to file name. The file naming convention is: STOP_Z₁-M₁-Z₂.DAT, with Z₁ and M₁ the nuclear charge and mass of the ion and Z₂ the nuclear charge of the target element. M₁ is rounded to the nearest integer value. Example: STOP_2_4_14.DAT should contain stopping power data for ⁴He in Si.
3. The file format is shown in Figure 3.19: The files can start with an arbitrary number of comment lines. These lines are ignored. The stopping power data are organized in two columns as function of energy, with the energy in keV and the stopping power in keV/1E15 at./cm². The data have to be ordered in ascending order. The data points may have irregular energy spacing, i.e. the energy step from one data point to the next may vary. The stopping power must be the total stopping power, i.e. the sum of electronic and nuclear stopping. SIMNRA uses linear interpolation between the data points.

3.20. Energy calibration issues

3.20.1. Detector nonlinearity

A solid state detector, which is used in most ion beam experiments, does not measure the real particle energy, but a somewhat smaller energy. This has been called pulse height defect. The pulse height defect is due to:

1. The energy and particle dependent energy loss in the top electrode and the dead layer of the semiconductor detector [36, 56, 105].
2. Only the electronic energy loss in the active detector region is measured, while nuclear energy loss is not detected [36]. The nuclear energy loss is energy and particle dependent.
3. Heavy ions produce a high density of electron-hole pairs. The electron-hole pairs may recombine before separation by the electric field in the detector. This has been called plasma effect [36]. The plasma effect is energy and particle dependent.

The energy dependence of the pulse height defect results in a nonlinearity of the energy calibration. The energy loss in the top electrode and the dead layer can be accounted in the following way:

- You can create a foil consisting of two layers in front of the detector. Layer 2 is composed of the material of the top electrode (usually Au) and has the thickness of the electrode (usually the thickness is supplied by the manufacturer of the detector). Layer 1 is composed of silicon in the case of a silicon detector and has the thickness of the dead layer (the dead layer is the insensitive region near the electrode). The dead layer thickness can be obtained only experimentally by tilting the detector.

For light ions (protons and He) this should be sufficient to achieve a linear energy calibration. But the nuclear energy loss and the plasma effect, which are both important for heavy ions, are not taken into account by this procedure.

An easier way to account for detector nonlinearities is to use a non-linear energy calibration with a quadratic correction term of the form

$$E \text{ [keV]} = A + B \times \text{channel} + C \times \text{channel}^2.$$

See [subsection 3.5.1](#) for details.

3.20.2. Energy calibration for different ion species

As already shown in [subsection 3.20.1](#) the pulse height defect depends on the particle species. This requires an individual energy calibration for each ion species if different ion species are detected, as is the case in NRA and ERDA measurements with incident heavy ions. SIMNRA offers the possibility to use an individual nonlinear energy calibration for each detected species, see [subsection 3.5.1](#) for details.

3.21. Data smoothing

SIMNRA provides a smoothed line through the experimental data points. This line is solely intended to guide the eye through the forest of data points in the case of strongly scattered data with low count statistics. This can be helpful for a visual judgment of the quality of a fit or for the identification of unresolved peaks.

The smoothed curve should *never* be shown alone, but always with the individual data points on the same graph, and quantitative evaluations should *never* be made on the basis of the smoothed curve. SIMNRA never uses the smoothed curve for any quantitative evaluations, such as fitting.

SIMNRA uses a smoother based on the well-known [4253H,twice] smoother, which has excellent low-pass transfer characteristics, negligible Gibbs rebound, and resistance to the effects of Gaussian and non-Gaussian disturbances [119].

The smoother can be called [(n-1)2n3H,twice] smoother (with window width n) and performs consecutively the following operations:

1. Running median with window width $n - 1$.
2. Running median with window width 2.
3. Running median with window width n .
4. Running median with window width 3.
5. Hanning, i.e. smoothing with a running weighted average with a windows width of 3 and triangular weights (0.25, 0.5, 0.25).
6. Roughing by computing the residuals between the original data and the smoothed curve. The residuals are then smoothed as above and added to the smoothed curve from the first pass.

This smoother preserves edges (such as high- or low-energy edges of high-Z element layers in RBS) without smearing them, which makes it very suitable for ion beam analysis data. However, peak areas are not necessarily preserved if n gets large: Peak areas of the smoothed curve therefore should not be used for further evaluations.

The smoothed curve is shown by clicking [Plot: Show Spectrum: Smoothed Experimental Spectrum](#) or by clicking [Tools: Smoothed Experimental Spectrum...](#), see subsection 3.9.4. The window width n can be changed by clicking [Tools: Smoothed Experimental Spectrum...](#), see subsection 3.9.4.

3.22. Programming support

3.22.1. Command line parameters

- */regserver* — Registers the OLE automation server in the Windows Registry and terminates the program. This is done automatically by the setup program, so there should be rarely, if ever, the necessity to use this parameter.
- */unregserver* — Unregisters the OLE automation server from the Windows Registry and terminates the program. This is done automatically by the uninstall program, so there should be rarely, if ever, the necessity to use this parameter.
- */register* — Writes default registry entries for SIMNRA (stopping power data path, cross-section data path, visible menu entries, creation of a backup file, etc., see [subsection 3.11.2](#)) and terminates the program. This is done automatically by the setup program, so there should be rarely, if ever, the necessity to use this parameter. OLE automation server entries are not created by */register*: These are created by */regserver*.
- */unregister* — Deletes all registry entries used by SIMNRA (except OLE automation server entries) and terminates the program. This is done automatically by the uninstall program, so there should be rarely, if ever, the necessity to use this parameter.
- */PIGE* — Sets SIMNRA into PIGE mode.
- Additionally SIMNRA accepts one optional command line parameter, which is the name with full path of a nra- or xnra-file. This file is opened upon startup. Use double quotes (") if the file name contains blanks.

Example: `simnra "c:\test.xnra"`

3.22.2. OLE automation

SIMNRA is an OLE 2.0 automation server, which allows other applications to control SIMNRA. This is useful for automatic processing of large numbers of spectra, and allows to implement additional functions, such as the calculation of spectra from laterally inhomogeneous samples [54]. The available OLE objects and methods are described in [Appendix A](#), some sample programs can be found in [section A.18](#).

3.23. The xnra file format

SIMNRA 6.50 and higher use the new *xnra* file format. Earlier versions used the meanwhile outdated *nra* file format. The old *nra* file format is no longer supported for writing, but still can be opened.

The *xnra* file format is an xml-based file format according to the IBA data format (IDF) definition [17]. While the IDF definition is very versatile and allows a large flexibility in data definition and storage, the *xnra* definition is more strict: Several items are mandatory, while the use of other items is more restricted than in general IDF files.

The following describes *xnra* 1.0.

1. *Required:* The SIMNRA name space must be defined for the whole document by the name space definition `xmlns:simnra="http://www.simnra.com/simnra"` in the IDF root node.
2. *Required:* The node `idf/attributes/simnra:filetye` must exist with the content *xnra*.
3. *Recommended:* A node `idf/attributes/simnra:xnraversionnr` with the content 1.0 may exist. This is not strictly required, but recommended.
4. *Required:* The file must contain exactly one sample. Multiple samples are not allowed.
5. *Required:* The file must contain at least one spectrum, which may be either an experimental or simulated spectrum.
6. *Required:* The file may not contain more than one experimental spectrum: Multiple experimental spectra are not allowed. If the experimental spectrum is missing, then a simulated spectrum must exist.
7. *Required:* The file may not contain more than one simulated spectrum: Multiple simulated spectra are not allowed. If the simulated spectrum is missing, then an experimental spectrum must exist.

An example for a valid *xnra* file (with most branches collapsed) is shown in [Figure 3.20](#).

3. Using SIMNRA

```
<?xml version="1.0" encoding="UTF-8" ?>
- <idf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://idf.schemas.itn.pt"
  xmlns:simnra="http://www.simnra.com/simnra">
+ <users>
+ <notes>
- <attributes>
  <idfversion>1.0</idfversion>
  <filename>test.xnra</filename>
  <createtime>2010-05-21 22:00:15</createtime>
  <simnra:simnraversionnr>6.50</simnra:simnraversionnr>
  <simnra:filetype>xnra</simnra:filetype>
  <simnra:xnraversionnr>1.0</simnra:xnraversionnr>
</attributes>
- <sample>
+ <users>
+ <notes>
+ <elementsandmolecules>
+ <structure>
+ <spectra>
</sample>
</idf>
```

Figure 3.20.: Example for a valid *xnra* file, with most branches collapsed.

4. Physics

During the last decades several programs for the simulation of backscattering spectra have been developed. The most common program is Doolittle's RUMP [40, 41]. However, RUMP uses several approximations to save computing time. The increase in computer power during the last years has made it possible to drop several of the approximations used by RUMP. SIMNRA offers more freedom in the use of non-Rutherford cross-sections and nuclear reactions, treats several topics such as straggling and convolution more precise and adds new possibilities such as dual scattering. This section describes the physics involved in the simulation of a backscattering spectrum as performed by SIMNRA.

The target is subdivided into shallow sublayers. Each simulated spectrum is made up of the superimposed contributions from each isotope of each sublayer of the sample target. The thickness of each sublayer is chosen in such a way that the energy loss in each sublayer is about the stepwidth of the incoming particles. When the incident particles penetrate a sublayer, they loose energy due to electronic and nuclear energy loss and the beam energy is spread due to straggling. The calculation of the energy loss is described in detail in [section 4.5](#), and the calculation of straggling in [section 4.8](#). SIMNRA calculates the energy of backscattered particles¹ from the front and the backside of the sublayer, and the energy of these particles when reaching the detector after passing to the target surface and traversing a foil in front of the detector, see [Figure 4.1](#). The contribution of each isotope in each sublayer will be referred to as a brick.

To account for energy straggling and the finite energy resolution of the detector the brick shown in [Figure 4.1](#) is convoluted with a Gaussian function $f(E, \sigma^2)$ with width

$$\sigma^2 = \sigma_{\text{Straggling Out}}^2 + \sigma_{\text{Detector}}^2. \quad (4.1)$$

$\sigma_{\text{Straggling Out}}^2$ is the variance of the energy distribution of the outgoing particles due to energy loss straggling, and $\sigma_{\text{Detector}}^2$ is the energy resolution of the detector.

The final contribution to the energy spectrum of each isotope in each sublayer is given by

$$S(E) = \int_0^\infty S_0(E') f(E', \sigma^2(E')) dE' \quad (4.2)$$

Here $S_0(E)$ is the energy spectrum before convolution and $S(E)$ the spectrum after the convolution. Note that the width of the Gaussian changes throughout the brick due to different straggling contributions.

¹A 'backscattered' particle may be a recoil or a product in a nuclear reaction as well.

4. Physics

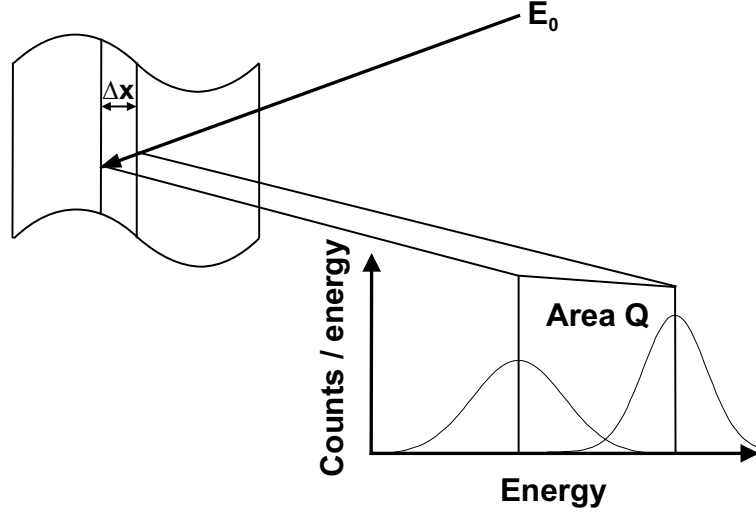


Figure 4.1.: Notation used for a single brick.

The Number of counts N_i in each channel i is given by integrating $S(E)$ over the channel width from the minimum to the maximum energy of each channel:

$$N_i = \int_{E_{min}(i)}^{E_{max}(i)} S(E') dE' \quad (4.3)$$

Equation 4.2 and Equation 4.3 can be put together into a 2-dimensional integral, which is computed by SIMNRA by means of a 2-dimensional Gauss-Legendre integration. The accuracy of the integration is about 10^{-4} .

The area Q of the brick in Figure 4.1 is calculated by using the integrated cross-section, see section 4.4, which is stable against fast varying structures in the cross-section, such as sharp resonances.

4.1. Atomic data

Masses of elements and isotopes used by SIMNRA 6.81 and higher have been taken from the recommended values of the 2012 update to the atomic mass evaluation [121]². Isotopic masses are stored in the file `ATOMDATA.DAT`. This file contains also the masses of some isotopes which do not occur naturally but may be created in nuclear reactions and are necessary for kinematic calculations.

Abundances of isotopes have been taken from the recommended values of isotopic abundances in [22]. Isotopic abundances are stored in the file `ATOMDATA.DAT`.

Values for fundamental physical constants were taken from the 2018 CODATA recommended values of the fundamental physical constants [112]³.

²Earlier versions of SIMNRA used the recommended values of the 1995 update [10].

³SIMNRA 7.02 and 7.03 used the CODATA 2014 recommended values of the fundamental physical constants [86], SIMNRA versions prior to 7.02 used the 2010 CODATA values [85].

4.2. Scattering kinematics

4.2.1. Elastic scattering

Energy of backscattered projectiles

The energy E_1 of a backscattered projectile with incident energy E_0 and mass M_1 after scattering is given in the laboratory system by

$$E_1 = E_0 \frac{M_1^2}{(M_1 + M_2)^2} \left\{ \cos \theta \pm \left[\left(\frac{M_2}{M_1} \right)^2 - \sin^2 \theta \right]^{1/2} \right\}^2 \quad (4.4)$$

θ is the scattering angle and M_2 the mass of the target nucleus initially at rest. For $M_1 < M_2$ only the plus sign in Equation 4.4 applies. If $M_1 > M_2$ then Equation 4.4 has two solutions, and the maximum possible scattering angle θ_{max} is given by

$$\theta_{max} = \arcsin \left(\frac{M_2}{M_1} \right). \quad (4.5)$$

The second solution for $M_1 > M_2$ is obtained because different impact parameters d result in the same scattering angle θ . Consider for example the scattering angle $\theta = 0$: This angle is obtained for a head-on collision with $d = 0$, but also for very large impact parameters $d \approx \infty$.

SIMNRA uses both solutions of Equation 4.4, if kinematically possible.

Energy of recoils

The energy E_2 of a recoil is given in the laboratory system by

$$E_2 = E_0 \frac{4M_1M_2}{(M_1 + M_2)^2} \cos^2 \theta. \quad (4.6)$$

E_0 is the energy of the incident projectile, M_1 the mass of the projectile, M_2 the mass of the target nucleus initially at rest and θ the recoil angle, with $0^\circ \leq \theta \leq 90^\circ$.

4.2.2. Nuclear reactions

For the calculation of nuclear reactions kinematics we use the quantities listed in Table 4.1.

We define the following quantities:

$$\begin{aligned} A_{13} &= \frac{M_1 M_3}{(M_1 + M_2)(M_3 + M_4)} \frac{E_1}{E_T} \\ A_{14} &= \frac{M_1 M_4}{(M_1 + M_2)(M_3 + M_4)} \frac{E_1}{E_T} \\ A_{23} &= \frac{M_2 M_3}{(M_1 + M_2)(M_3 + M_4)} \left(1 + \frac{M_1}{M_2} \frac{Q}{E_T} \right) \\ A_{24} &= \frac{M_2 M_4}{(M_1 + M_2)(M_3 + M_4)} \left(1 + \frac{M_1}{M_2} \frac{Q}{E_T} \right) \end{aligned}$$

Table 4.1.: Quantities used for the calculation of nuclear reactions kinematics. The target nucleus is initially at rest. For exotherm reactions $Q > 0$, for endotherm reactions $Q < 0$.

	Mass	Energy
Incident ion	M_1	E_1
Target nucleus	M_2	0
Light product	M_3	E_3
Heavy product	M_4	E_4
Energy released in reaction		Q
Total energy		$E_T = E_1 + Q = E_3 + E_4$

The energy E_3 of the light product created in the nuclear reaction is then given in the laboratory system by

$$E_3 = E_T A_{13} \left[\cos \theta \pm \left(\frac{A_{24}}{A_{13}} - \sin^2 \theta \right)^{1/2} \right]^2 \quad (4.7)$$

θ is the emission angle of the light product in the laboratory system. For $A_{13} < A_{24}$ only the plus sign in Equation 4.7 applies. If $A_{13} > A_{24}$ then Equation 4.7 has two solutions, and the maximum possible emission angle θ_{max} of the light product is

$$\theta_{max} = \arcsin \left(\frac{A_{24}}{A_{13}} \right)^{1/2} \quad (4.8)$$

The energy E_4 of the heavy product created in the nuclear reaction is given in the laboratory system by

$$E_4 = E_T A_{14} \left[\cos \Phi \pm \left(\frac{A_{23}}{A_{14}} - \sin^2 \Phi \right)^{1/2} \right]^2 \quad (4.9)$$

Φ is the emission angle of the heavy product in the laboratory system. For $A_{14} < A_{23}$ only the plus sign in Equation 4.9 applies. If $A_{14} > A_{23}$ then Equation 4.9 has two solutions, and the maximum possible emission angle Φ_{max} of the heavy product is

$$\Phi_{max} = \arcsin \left(\frac{A_{23}}{A_{14}} \right)^{1/2} \quad (4.10)$$

SIMNRA uses Equation 4.7 and Equation 4.9 only with the plus sign. The second solution is neglected.

4.3. Cross-section data

4.3.1. Rutherford cross-section

Scattering cross-section

The Rutherford cross-section for backscattering is given in the laboratory system by

$$\sigma_R [\text{mb/sr}] = 5.1837436 \times 10^6 \left(\frac{Z_1 Z_2}{E [\text{keV}]} \right)^2 \frac{\left\{ (M_2^2 - M_1^2 \sin^2 \theta)^{1/2} + M_2 \cos \theta \right\}^2}{M_2 \sin^4 \theta (M_2^2 - M_1^2 \sin^2 \theta)^{1/2}} \quad (4.11)$$

θ is the scattering angle, Z_1 and M_1 are the nuclear charge and the mass of the projectile, respectively, and Z_2 and M_2 are the nuclear charge and the mass of the target atom, respectively. σ_R is the differential cross-section in the laboratory system⁴.

Recoil cross-section

The Rutherford cross-section for recoils is given in the laboratory system by

$$\sigma_R^{ERD} [\text{mb/sr}] = 2.0731 \times 10^7 \frac{[Z_1 Z_2 (M_1 + M_2)]^2}{(2M_2 E [\text{keV}])^2 \cos^3 \theta} \quad (4.12)$$

θ is the recoil angle in the lab system.

4.3.2. Screened Rutherford cross-sections

Experimental measurements indicate that actual cross-sections deviate from Rutherford at both high and low energies for all projectile-target pairs. The low-energy departures are caused by partial screening of the nuclear charges by the electron shells surrounding both nuclei [5, 52, 66, 110]. This screening is taken into account by a correction factor F :

$$\sigma = F \sigma_R$$

L'Ecuyer screening

Scattering cross-section For $\theta > 90^\circ$ the correction factor by L'Ecuyer *et al.* [66] is easy to use:

$$F_{L'Ecuyer} = 1 - \frac{0.04873 Z_1 Z_2^{4/3}}{E_{CM}} \quad (4.13)$$

E_{CM} is the energy in the center of mass system (in keV). Tabulated values of $F_{L'Ecuyer}$ can be found for example in [110]. But note, that L'Ecuyer used the factor 0.049 instead of the correct

⁴For $M_1 > M_2$ there may exist two different solutions of the kinematic equation Equation 4.4. The cross-section given by Equation 4.11 applies for the solution with the plus sign. The cross-section for the second solution of Equation 4.4 is obtained by calculating the scattering angle in the centre-of-mass system θ_{CM} from $\theta_{CM} = \theta - \arcsin[M_1/M_2 \sin \theta] + \pi$, calculating the Rutherford backscattering cross-section in the centre-of-mass system and transforming into the laboratory system.

0.04873 [68] in his original article. The correction for backscattering angles $\theta > 90^\circ$ at typical energies used in ion beam analysis usually is small. For 1 MeV ^4He ions on gold the correction is only about 3.5%.

L'Ecuyer screening makes only sense for light projectiles scattered from a heavier target atom at backward angles. L'Ecuyer screening cannot be used for recoils. At low energies and/or small scattering angles Andersen screening (subsubsection 4.3.2) or Universal screening (subsubsection 4.3.2) are more accurate.

Andersen screening

Scattering cross-section The correction factor by L'Ecuyer (Equation 4.13) is a first order correction and does not take into account the influence of the scattering angle θ . For $\theta < 90^\circ$ Equation 4.13 will underestimate the necessary correction to the Rutherford cross-section. A more accurate approximation is the angular- and energy dependent correction factor by Andersen *et al.* [5]:

$$F_{\text{Andersen}} = \frac{\left(1 + \frac{1}{2} \frac{V_1}{E_{CM}}\right)^2}{\left\{1 + \frac{V_1}{E_{CM}} + \left[\frac{V_1}{2E_{CM} \sin \theta_{CM}/2}\right]^2\right\}^2} \quad (4.14)$$

θ_{CM} is the scattering angle in the center of mass system. The increase in the kinetic energy V_1 is given by

$$V_1 [\text{keV}] = 0.04873 Z_1 Z_2 \left(Z_1^{2/3} + Z_2^{2/3}\right)^{1/2}$$

The dependence of the correction factor F_{Andersen} from the scattering angle θ for ^4He scattered from gold is shown in Figure 4.2 for different ^4He energies. Dashed lines are the angular independent correction factor by L'Ecuyer. For large scattering angles the correction factors by L'Ecuyer and Andersen are near to unity and similar, however, for small scattering angles the correction by Andersen becomes large and the angle-independent L'Ecuyer correction underestimates the deviations from the Rutherford cross-section.

The Andersen correction was developed for the weakly screened region, i.e. for sufficiently small impact parameters at sufficiently high energies and not too small scattering angles. The Andersen correction is usually sufficiently accurate at energies and angles used in ion beam analysis, but may be inaccurate at energies below a few hundred keV and/or small scattering angles. At low energies and/or small scattering angles Universal screening (subsubsection 4.3.2) is more accurate.

Recoil cross-section The Andersen screening correction from Equation 4.14 was developed for scattered projectiles. The screened recoil cross section $\sigma_R(E_R, \phi)$ at incident projectile energy E_R and recoil angle ϕ can be obtained from the screened scattering cross-section $\sigma_S(E_S, \theta)$ at incident energy E_S and scattering angle θ using inverse scattering, see for example [58].

$$\sigma_R(\phi) = 4 \frac{\sin^2 \theta \cos(\theta_c - \theta) \cos \phi}{\sin^2 \theta_c} \sigma_S(\theta) \quad (4.15)$$

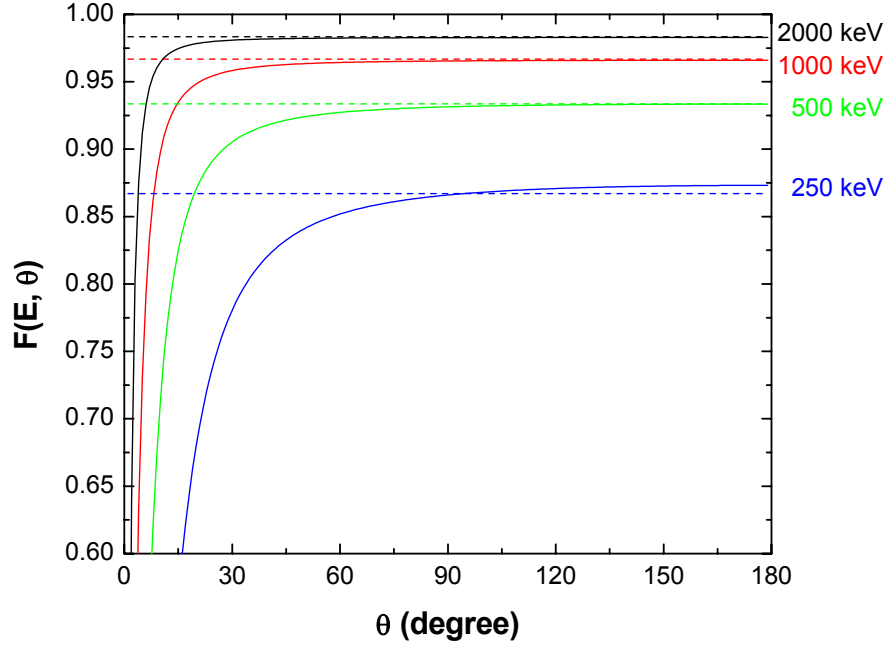


Figure 4.2.: Angular dependence of the correction factors for the Rutherford cross-section by L'Ecuyer (Equation 4.13, dashed lines) and Andersen (Equation 4.14, solid lines) for ^4He backscattered from gold at different energies.

with the center-of-mass angle θ_c

$$\theta_c = \pi - 2\phi = \theta + \arcsin\left(\frac{M_2}{M_1} \sin \theta\right) \quad (4.16)$$

and the matching energies

$$E_S = \frac{M_2}{M_1} E_R. \quad (4.17)$$

M_1 is the projectile and M_2 the target mass.

Universal screening

The L'Ecuyer (subsubsection 4.3.2) and Andersen (subsubsection 4.3.2) screening corrections were developed for the weakly screened region, i.e. for sufficiently small impact parameters at sufficiently high energies and not too small scattering angles. At low energies and/or small scattering angles the Universal screening is more accurate.

Scattering cross-section The Universal screening function is obtained from the Universal potential [136]:

$$V_U(r) = \frac{Z_1 Z_2 e^2}{a(r/a)} \left(0.1818^{-3.2r/a} + 0.5099e^{-0.94230r/a} + 0.2802e^{-0.4029r/a} + 0.02817e^{-0.2016r/a} \right), \quad (4.18)$$

with the Universal screening length

$$a = \frac{0.8853a_B}{Z_1^{0.23} + Z_2^{0.23}}. \quad (4.19)$$

a_B is the Bohr radius, Z_1 and Z_2 the nuclear charges of projectile and target atom.

By introducing the factor C , with

$$C = \frac{aM_2E}{Z_1Z_2e^2(M_1 + M_2)} \quad (4.20)$$

the Universal cross-section can be tabulated independent of a specific ion-target combination. M_1 and M_2 are the masses of projectile and target atom, E the projectile energy, and e^2 the square of the nuclear charge, with $e^2 \approx 14.4$ eVÅ.

In order to obtain the cross-section from the potential the classical scattering integral has to be solved numerically. This has been done in [96] using the Gauß-Mehler integration scheme with 10000 nodes, thus providing a very high level of accuracy. The calculated cross-section values were published as a technical report. The cross-section data were obtained by scanning and digitizing the tables from [96]. Quadratic interpolation between the logarithmic energy values $\ln(E_i)$ and linear interpolation between the scattering angles θ_j is used, where E_i , θ_j and $\sigma(E_i, \theta_j)$ are the tabulated values for energy E_i , scattering angle θ_j and cross-section σ at E_i, θ_j .

L'Ecuyer, Andersen and Universal screening are compared in Figure 4.3 for ^4He backscattered from gold at 165° . At 500 keV Universal and Andersen/L'Ecuyer screening deviate by about 2%. At lower energies the deviations get larger, and Andersen/L'Ecuyer screening get more and more inaccurate.

Recoil cross-section The recoil cross-section with Universal screening is obtained using inverse scattering, see paragraph 4.3.2.

4.3.3. Non-Rutherford cross-sections

High energy deviations

At high energies the cross-sections deviate from Rutherford due to the influence of the nuclear force. A useful formula above which energy E_{NR} deviations from Rutherford can be expected was given by Bozoian [27–29]:

$$\begin{aligned} E_{NR} [\text{MeV}] &= \frac{M_1 + M_2}{M_2} \frac{Z_2}{10} \quad \text{for } Z_1 = 1 \\ E_{NR} [\text{MeV}] &= \frac{M_1 + M_2}{M_2} \frac{Z_1 Z_2}{8} \quad \text{for } Z_1 > 1 \end{aligned}$$

E_{NR} is the energy at which the deviation from the Rutherford cross-section gets $> 4\%$.

SIMNRA does not check if the cross-sections at a given energy are Rutherford or not. It is in the responsibility of the user to choose the correct cross-sections. The above formulas may be useful to estimate if the cross-section is still Rutherford or not.

For non-Rutherford cross-sections SIMNRA uses differential cross-section data from various sources. The use of non-Rutherford cross-sections is described in full detail in section 3.7.

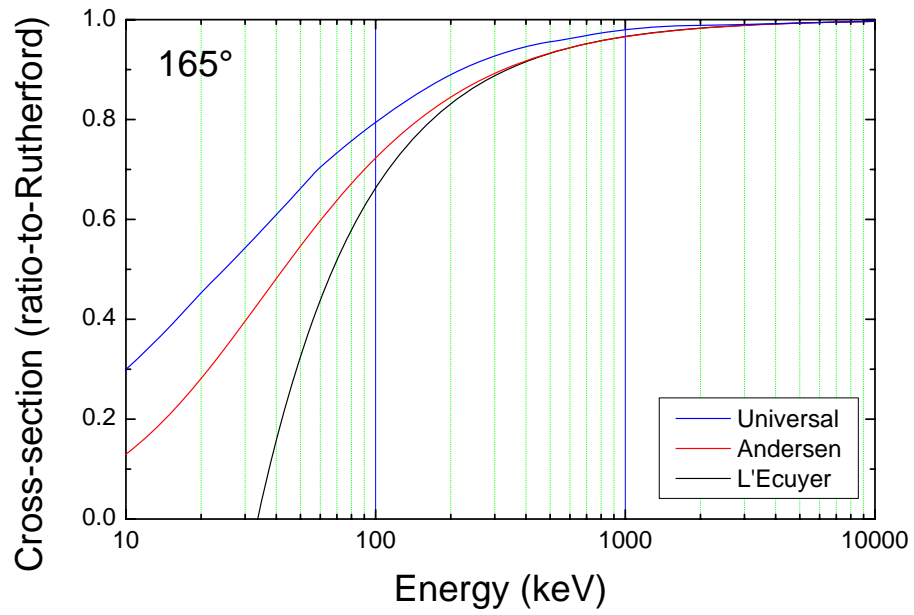


Figure 4.3.: Energy dependence of the screening factors to the Rutherford cross-section by L'Ecuyer, Andersen and Universal screening for ^4He backscattered from gold at 165° .

Mott scattering

The scattering of identical particles in forward direction (such as ^4He on ^4He , ^{12}C on ^{12}C , ^{28}Si on ^{28}Si etc.) results in a quantum-mechanical interference term and deviates from Rutherford scattering already at low energies [31, 32]. This is called Mott scattering and is shown in Figure 4.4. The Rutherford cross-section is a linear superposition of the Rutherford scattering and recoil cross-sections from Equation 4.11 and Equation 4.12. The experimental data deviate significantly from the Rutherford prediction and have to be described by the Mott cross-section. This case may occur in heavy ion ERDA, if the incident ion is present as element in the sample.

The program *MottCalc* can be used for calculating Mott scattering cross-sections, see subsection 3.12.4 for details.

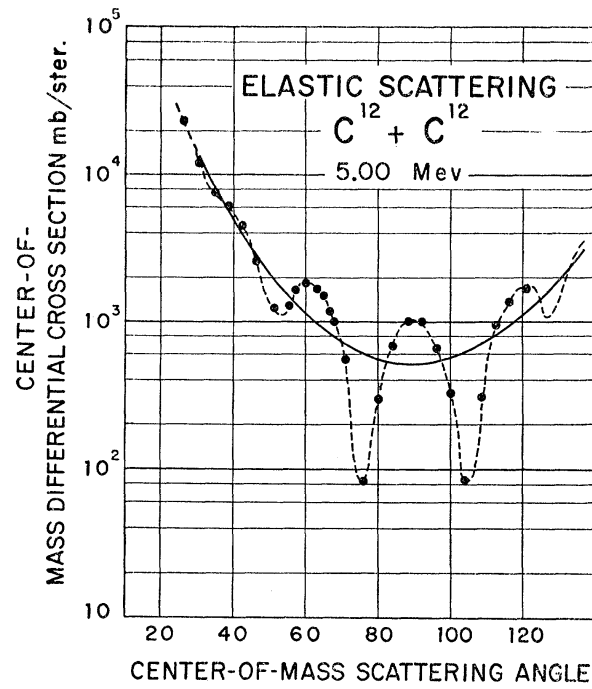


Figure 4.4.: Angular distribution for elastic scattering of ^{12}C on ^{12}C at a center-of-mass energy of 5 MeV. The solid curve is the Rutherford prediction; the dashed curve is the Mott prediction. Dots are experimental data. From [32].

4.4. Number of backscattered particles

The number of backscattered particles Q (i.e. the area of the brick in [Figure 4.1](#)) from a thin layer with thickness Δx can be calculated by using the cross-section at the mean energy \bar{E} in the layer:

$$Q = \frac{N\Delta\Omega}{\cos \alpha} \sigma(\bar{E}) \Delta x, \quad (4.21)$$

with $N\Delta\Omega$ the number of incident particles times the solid angle of the detector, $\sigma(\bar{E})$ the differential cross-section evaluated at the mean energy \bar{E} , and α the angle of incidence. SIMNRA 5.01 and earlier used [Equation 4.21](#) for the calculation of the number of backscattered particles. This is, however, only valid if the layer is sufficiently thin, i.e. if the energy loss in the layer is small enough, so that the cross-section does not change strongly. If the cross-section has structures such as sharp resonances, this requires very thin layers, which slows down the calculation considerably.

A more accurate approximation than [Equation 4.21](#) is obtained by integrating [Equation 4.21](#):

$$Q = \frac{N\Delta\Omega}{\cos \alpha} \int_{x_0}^{x_1} \sigma(E(x)) dx \quad (4.22)$$

$$= \frac{N\Delta\Omega}{\cos \alpha} \int_{E_0}^{E_1} \sigma(E) \frac{dx}{dE} dE \quad (4.23)$$

with x_0 and x_1 the start and end depths of the thin layer, and $E_0 = E(x_0)$, $E_1 = E(x_1)$ the corresponding energies. By assuming a constant stopping power $S = dE/dx(\bar{E})$ throughout the layer, we get $E(x) = E_0 - Sx$, and

$$Q = \frac{N\Delta\Omega}{\cos \alpha} \frac{1}{S} \int_{E_1}^{E_0} \sigma(E) dE. \quad (4.24)$$

[Equation 4.24](#) is used by SIMNRA 5.02 and higher, providing a higher accuracy and better stability than [Equation 4.21](#) in the case of narrow structures in the cross section. While [Equation 4.21](#) neglects all stopping power effects, [Equation 4.24](#) takes a constant stopping power into account, neglecting effects due to the variation of the stopping power. [Equation 4.24](#) is used if *Weighting of cross-sections by straggling* is set to *Fast* in *Setup:Calculation:Parameter*, see [section 3.5.3](#).

However, [Equation 4.24](#) still neglects the effects of straggling. After traversing some thickness of material the energy distribution of an ensemble of particles gets broader due to the various straggling effects (electronic energy-loss straggling, multiple scattering, porosity, etc.), see [Figure 4.1](#) for a schematic illustration. The energy distribution of an ensemble of straggled particles can be described by a distribution function $f(E)$, where $f(E)$ is typically close to Gaussian and normalized to $\int f(E) dE = 1$. A mean cross-section $\bar{\sigma}$ for this energy distribution then can be defined by

$$\bar{\sigma} = \int f(E) \sigma(E) dE, \quad (4.25)$$

with $\sigma(E)$ the energy-dependent scattering cross-section.

SIMNRA 6 and earlier used the simple approximation $\bar{\sigma} = \sigma(\bar{E})$, with \bar{E} the mean energy of the energy distribution: $\bar{E} = \int E f(E) dE$. This allows a very fast computation and is sufficiently

accurate for slowly varying cross-sections such as Rutherford cross-section. As has been shown in [15], however, this approximation gets inaccurate for cross-sections with structure (such as resonant cross-sections for scattering of nuclear reactions), if the energy spread of the straggling distribution gets comparable to the width of structures in the cross-section. SIMNRA 6.72 and higher evaluates Equation 4.25 by numerical integration.

Moreover, the shape of the cross-section influences not only the number of scattered particles via the mean cross-section, but influences also the shape of the energy distribution of the ensemble of particles after scattering. If a particle had the energy E before scattering, then the energy E' after scattering is given by $E' = KE$, with K the kinematic factor. For a constant (or slowly varying) cross-section the energy distribution after scattering $f'(E')$ is then simply given by $f'(E') = 1/Kf(E)$, which has mean value $\bar{E}' = K\bar{E}$, variance $V' = K^2V$ and third central moment $M' = K^3M$, with V the variance and M the third central moment of the energy distribution before scattering. For a cross-section with structure this has to be modified to $f'(E')dE' = 1/\bar{\sigma}f(E)\sigma(E)dE$, i.e. the shape of the energy distribution after scattering is influenced by the energy dependence of the cross-section: A Gaussian energy distribution before scattering will usually become non-Gaussian after scattering, if the cross-section has narrow structures. The first three moments of the energy distribution after scattering are then given by

$$\bar{E}' = K/\bar{\sigma} \int Ef(E)\sigma(E)dE \quad (4.26)$$

$$V' = K^2/\bar{\sigma} \int (E - \bar{E})^2 f(E)\sigma(E)dE \quad (4.27)$$

$$M' = K^3/\bar{\sigma} \int (E - \bar{E})^3 f(E)\sigma(E)dE. \quad (4.28)$$

SIMNRA 6.72 and higher uses numerical integration in order to calculate the first three moments of the energy distribution after scattering using Equation 4.26 to Equation 4.28. The energy distribution is then propagated to the surface. This procedure takes the first three moments of the energy distributions into account and requires to set *Weighting of cross-sections by straggling to Accurate* in *Setup:Calculation:Parameter*, see section 3.5.3.

This change of the shape of the energy distribution by the scattering cross-section has been already pointed out in [15]. However, the procedure described in [15] is somewhat inconsequent, because the shape of the energy distribution is assumed to be always Gaussian, and only the mean cross-section according to Equation 4.25 and the change of the mean energy according to Equation 4.26 is taken into account.

The energy spectrum of 2000 keV protons backscattered from pyrolytic graphite is shown in Figure 4.5. The spectrum is dominated by the well-known resonance in the cross-section at about 1740 keV. The simulation uses the SigmaCalc cross-section from [50]. The simulation with SIMNRA 6 gives a resonance peak, which is too narrow and too high, while SIMNRA 7 perfectly reproduces the shape of the peak. It should be noted, though, that for bulk samples SIMNRA 6 still yields the correct total number of counts (i.e. the integrated number of counts for the peak is almost identical for SIMNRA 6 and SIMNRA 7), but the width of the simulated peak is usually too narrow.

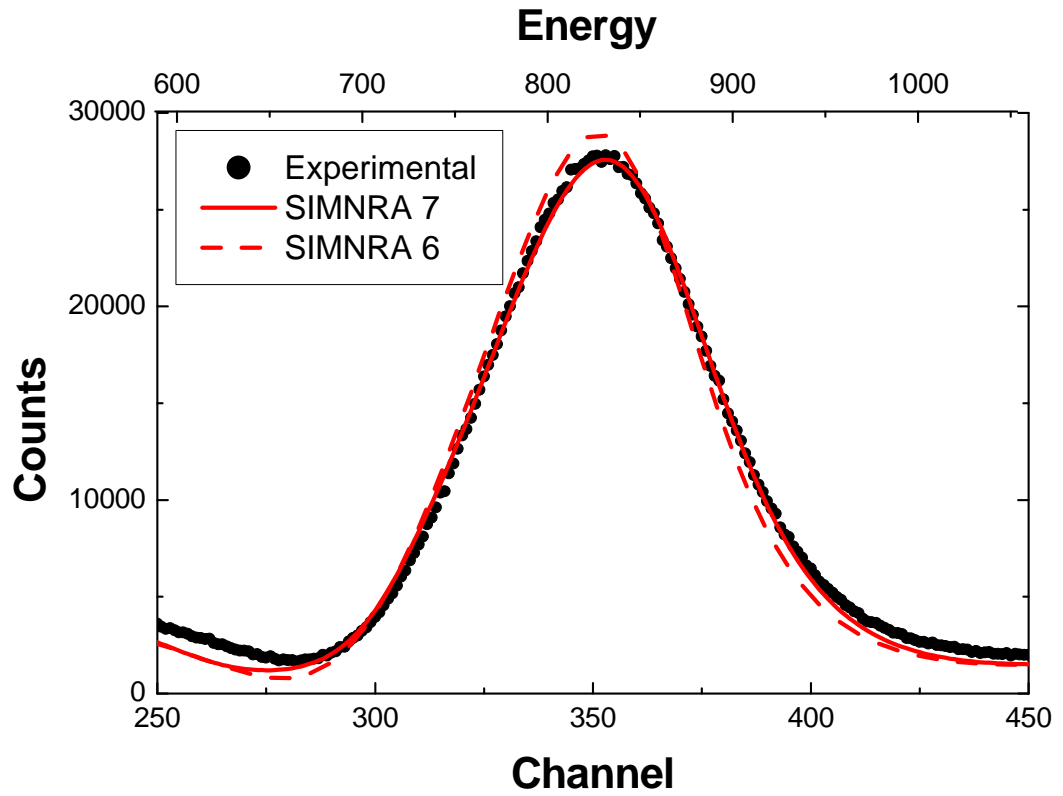


Figure 4.5.: Backscattering of 2000 keV protons from pyrolytic graphite at a scattering angle of 165° . Dots: Experimental data; Solid line: Simulation with SIMNRA 7 using the improved treatment of resonant cross-sections; Dashed line: Simulation using SIMNRA 6. The SigmaCalc cross-section from [50] was used for both simulations.

4.5. Evaluation of energy loss

The energy E of a particle after passing through a layer of material with thickness x is given by the integral equation

$$E(x) = E_0 - \int_0^x \frac{dE}{dx'}(E(x'), x') dx' \quad (4.29)$$

Here we assume that the particle starts with initial energy E_0 at the surface ($x = 0$), and $dE/dx'(E(x'), x')$ is the energy and depth dependent stopping power. x is the path length into the material, measured in areal density (10^{15} atoms/cm²). By differentiating Equation 4.29, we obtain the differential equation

$$\frac{dE}{dx} = -\epsilon(E), \quad (4.30)$$

where this is the defining equation for $\epsilon(E)$, the energy dependent stopping cross-section.

All SIMNRA versions before 5.70 used the algorithm of Doolittle for the evaluation of energy loss, initially developed for the RUMP program [40]. Doolittle expands the particle energy into a Taylor series, and the energy E_1 after a layer of material with thickness Δx , is given by

$$E_1 = E_0 + \Delta x \frac{dE}{dx}(E_0) + \frac{1}{2} \Delta x^2 \frac{d^2E}{dx^2}(E_0) + \frac{1}{6} \Delta x^3 \frac{d^3E}{dx^3}(E_0) \quad (4.31)$$

where $\epsilon' = d\epsilon/dE$ is the first and $\epsilon'' = d^2\epsilon/dE^2$ the second derivative of ϵ . ϵ , ϵ' and ϵ'' are evaluated at the incident energy E_0 . The terms in Equation 4.31 look as follows:

$$\frac{dE}{dx} = -\epsilon \quad (4.32)$$

$$\frac{d^2E}{dx^2} = \frac{d}{dx}(-\epsilon) = -\frac{d\epsilon}{dE} \frac{dE}{dx} = \epsilon' \epsilon \quad (4.33)$$

$$\frac{d^3E}{dx^3} = \frac{d}{dx}(\epsilon' \epsilon) = \frac{d\epsilon'}{dE} \epsilon + \epsilon' \frac{d\epsilon}{dE} = -\epsilon'' \epsilon^2 - \epsilon'^2 \epsilon \quad (4.34)$$

With ϵ , ϵ' and ϵ'' evaluated at E_0 . This gives the final result

$$E_1 = E_0 - \Delta x \epsilon + \frac{1}{2} \Delta x^2 \epsilon \epsilon' - \frac{1}{6} \Delta x^3 (\epsilon'' \epsilon^2 + \epsilon'^2 \epsilon) \quad (4.35)$$

ϵ' and ϵ'' were calculated by SIMNRA by numerical differentiation of the stopping power data.

Major disadvantage of Equation 4.35 is the need of the first and second derivative of the stopping power. Stopping powers supplied by the SRIM program don't have smooth derivatives, and the use of Equation 4.35 can result in large errors (see below). Moreover, Equation 4.35 does not provide a criterion how to select the step width Δx in order to obtain a desired accuracy, and it does not provide an error estimate.

Therefore Equation 4.35 was replaced by a more accurate and stable Runge-Kutta algorithm in SIMNRA 5.70 [92]. Equation 4.30 is a first order differential equation with initial values $E(x=0) = E_0$ and $dE/dx(x=0) = -\epsilon(E_0)$. SIMNRA uses a fifth order Runge-Kutta method with embedded fourth order Runge-Kutta with Cash-Karp parameters for an error estimate and automatic step width control [92]. The error is some eV for energy losses of the order of 1 MeV.

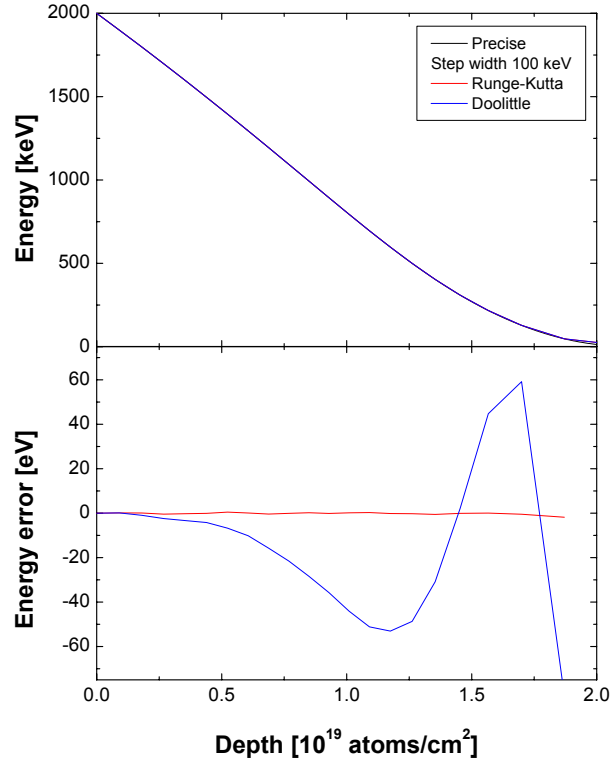


Figure 4.6.: Slowing down of 2 MeV ^4He in Au with Ziegler/Biersack stopping, calculated with different numerical methods. Top: ^4He energy as a function of depth. Bottom: Error of the calculated energy. The 'Precise' result was obtained with a Runge-Kutta method using a step width of 1 keV.

The energy loss of ^4He ions in Au and the numerical error of the energy loss calculation are shown in Figure 4.6 for Ziegler/Biersack stopping. The Ziegler/Biersack stopping, as used by SIMNRA (see subsection 4.6.2), is smooth with continuous second derivative due to spline interpolation of the input data, and the energy loss calculation according to Doolittle's formula (Equation 4.35) results in a maximum error of about 50 eV. Runge-Kutta with identical step width is more accurate, with a maximum error below 1 eV. The same calculation, but with SRIM 2003 stopping powers, is shown in Figure 4.7. The SRIM 2003 stopping power has no continuous derivative, and Doolittle's method results in serious deviations. The Runge-Kutta method still gives accurate results with a maximum error of about 20 eV.

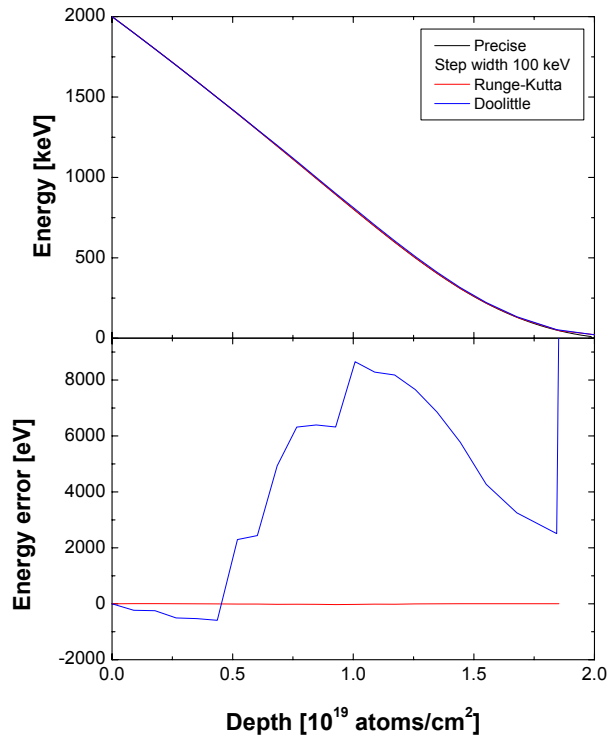


Figure 4.7.: Slowing down of 2 MeV ^4He in Au with SRIM 2003 stopping, calculated with different numerical methods. Top: ^4He energy as a function of depth. Bottom: Error of the calculated energy. The 'Precise' result was obtained with a Runge-Kutta method using a step width of 1 keV.

4.6. Stopping power data

SIMNRA offers the possibility to use several different sets of electronic stopping power data for the stopping of light and heavy ions in all elements. The differences between the data sets are typically < 5%, but may be larger in some cases, especially for ions other than H or He. See [69, 89, 90, 134] for a discussion of the accuracy of stopping powers.

4.6.1. Andersen-Ziegler stopping

Hydrogen

If Andersen-Ziegler stopping is selected SIMNRA uses the electronic stopping power data by Andersen and Ziegler [6] for the stopping of incident protons, deuterons and tritons in all elements. The electronic stopping power S_e in eV/(10^{15} atoms/cm²) for an incident hydrogen ion with energy/mass E in keV/amu is given by

$$\frac{1}{S_e} = \frac{1}{S_{Low}} + \frac{1}{S_{High}} \quad (4.36)$$

with

$$S_{Low} = A_2 E^{0.45} \quad (4.37)$$

and

$$S_{High} = \frac{A_3}{E} \ln \left[1 + \frac{A_4}{E} + A_5 E \right] \quad (4.38)$$

$A_2 - A_5$ are fitting coefficients and tabulated in [6]. They are stored in the file STOPH.DAT, see Table 2.1. Equation 4.36 to Equation 4.38 are valid for $10 \text{ keV} \leq E < 1 \text{ MeV}$. For energies in the range $1 \text{ MeV} - 100 \text{ MeV}$ the electronic stopping power S_e is given by

$$S_e = \frac{A_6}{\beta^2} \left[\ln \frac{A_7 \beta^2}{1 - \beta^2} - \beta^2 - \sum_{i=0}^4 A_{i+8} (\ln E)^i \right] \quad (4.39)$$

$A_6 - A_{12}$ are tabulated in [6], $\beta = v/c$, with v the ion velocity and c the speed of light. Equation 4.39 is used only if the switch *High energy stopping* in the *Setup: Calculation: Parameter* form is checked, see section 3.5.3. If unchecked, the program will use Equation 4.36 to Equation 4.38 at all energies. The program default is checked. The difference between Equation 4.36 and Equation 4.39 is small in most cases. The main problem using Equation 4.39 is that the first and second derivatives of Equation 4.36 and Equation 4.39 do not fit smoothly together at 1 MeV/amu . This may result in the appearance of kinks in the spectrum.

Helium

If Andersen-Ziegler stopping is selected then for incident ^3He and ^4He ions the electronic stopping power data by Ziegler [131] are used for all elements. The electronic stopping power S_e in eV/(10^{15} atoms/cm²) for incident ^4He ions with energy E in keV is given by

$$\frac{1}{S_e} = \frac{1}{S_{Low}} + \frac{1}{S_{High}} \quad (4.40)$$

with

$$S_{Low} = A_1 E^{A_2} \quad (4.41)$$

and

$$S_{High} = \frac{A_3}{E} \ln \left[1 + \frac{A_4}{E} + A_5 E \right] \quad (4.42)$$

$A_1 - A_5$ are fitting coefficients and tabulated in [131]. They are stored in the file `STOPHE.DAT`, see Table 2.1. Equation 4.40 to Equation 4.42 are valid for $1 \text{ keV} \leq E < 10 \text{ MeV}$. ^4He -stopping at higher energies (above 10 MeV) is not implemented in the program.

The stopping power of ^3He is identical to the stopping power of ^4He at the same velocity [131]. The stopping power of ^3He with energy E is obtained by taking the stopping power value at the energy $E(^4\text{He}) = 4/3 E(^3\text{He})$. The stopping power for ^3He is valid in the energy range $10 \text{ keV} \leq E < 7.5 \text{ MeV}$.

Heavy ions

The electronic stopping power of heavy ions in all elements is derived from the stopping power of protons using Brandt-Kitagawa theory [136, 137]. The formalism is described in detail in ref. [136], a short overview is given in [137]. The screening length Λ (eq. 3-29 of ref. [136]) is multiplied by an empirical correction factor which has been digitised from fig. 3-25 of ref. [136]. The correction factor for all elements is stored in the file `LCORRHI.DAT`, see Table 2.1.

Note that the switch *High energy stopping* in the *Setup: Calculation* menu has influence on the calculation of the stopping power for heavy ions with incident energies above 1 MeV/amu.

4.6.2. Ziegler-Biersack stopping

Hydrogen

If Ziegler-Biersack stopping is selected SIMNRA uses the electronic stopping power data by Ziegler, Biersack and Littmark [136] for the stopping of incident protons, deuterons and tritons in all elements. The electronic stopping power S_e in $\text{eV}/(10^{15} \text{ atoms}/\text{cm}^2)$ for an incident hydrogen ion with energy/mass E in keV/amu is given by

$$S_e = \frac{S_{Low} S_{High}}{S_{Low} + S_{High}} \quad (4.43)$$

with

$$S_{Low} = C_1 E^{C_2} + C_3 E^{C_4} \quad (4.44)$$

and

$$S_{High} = \frac{C_5}{E^{C_6}} \ln \left(\frac{C_7}{E} + C_8 E \right) \quad (4.45)$$

$C_1 - C_8$ are fitting coefficients and partly tabulated in [137]. They are stored in the file `SCOE.F.95A` (see Table 2.1) for all elements. Equation 4.43 to Equation 4.45 are valid in the

4. Physics

energy range $10 \text{ keV/amu} \leq E < 10 \text{ MeV/amu}$. For energies in the range 10–100 MeV/amu the electronic stopping power S_e is given by

$$S_e = C_9 + C_{10}x + C_{11}x^2 + \frac{C_{12}}{x} \quad (4.46)$$

with $x = \ln(E)/E$. $C_9 - C_{12}$ are fitting coefficients and tabulated in the file SC0EF.95B, see Table 2.1.

For energies below 10 keV/amu the electronic stopping power S_e is given by

$$S_e(E) = S_e(10) \left(\frac{E}{10} \right)^y \quad (4.47)$$

where $S_e(10)$ is the stopping power at 10 keV/amu and $y = 0.45$ for $Z_2 > 6$ and $y = 0.35$ for $Z_2 \leq 6$.

Helium

If Ziegler-Biersack stopping is selected then for incident ^3He and ^4He ions the electronic stopping power data by Ziegler, Biersack and Littmark [136] are used for all elements.

The electronic stopping of He ions in elements S_e is derived from the stopping power of protons for the same velocity S_p by using [136, 137]

$$S_e = S_p (\gamma_{He} Z_{He})^2 \quad (4.48)$$

Z_{He} is the helium charge and γ_{He} can be obtained from the simple polynomial fit

$$\gamma_{He}^2 = 1 - \exp \left[- \sum_{i=0}^5 C_i B^i \right] \quad (4.49)$$

with $B = \ln(E)$ and E in keV/amu. The coefficients C_i tabulated in [137] are incorrect, the correct coefficients (as taken from the SRIM 95 source code) are 0.2865, 0.1266, -0.001429, 0.02402, -0.01135, 0.001475.

Heavy ions

The electronic stopping power of heavy ions in all elements is derived from the stopping power of protons using Brandt-Kitagawa theory [136, 137]. The formalism is described in detail in ref. [136], a short overview is given in [137]. Fermi velocities of all target elements are stored in the file SC0EF.95A, a small correction to the Fermi velocity in the file SC0EF.95B. The ion's screening length Λ as a function of fractional charge is stored in the file SC0EF.95B. See Table 2.1 for the directories of these files.

Differences between SIMNRA and SRIM 97

The program SRIM by J. Ziegler is a widely used program for stopping power and range calculations. If Ziegler-Biersack stopping is selected SIMNRA uses the same stopping power routines and input data as SRIM 97 for elemental targets. The routines for stopping power calculations have been taken from the SRIM source code, the input data have been taken from the SRIM distribution. For incident hydrogen and helium ions SIMNRA therefore uses exactly the same electronic and nuclear stopping powers as the SRIM program. For incident heavy ions, however, there are *small* differences between SIMNRA and SRIM. SRIM uses linear interpolation for some input data, resulting in stopping powers with a noncontinuous derivative. SIMNRA uses spline interpolation instead of linear interpolation, resulting in stopping powers with continuous second derivative. The stopping powers calculated by SIMNRA are therefore more smooth than the original stopping powers calculated by SRIM, the differences are typically below 1%.

4.6.3. KKK stopping

As was shown by Konac et al. [61, 62], the SRIM 97 stopping power values (which are used by SIMNRA if Ziegler/Biersack stopping is selected) are somewhat inaccurate for the stopping of ions in C and Si. Konac et al. fitted their experimental values with the formulas

$$S_e(E) = E^s \ln(e + \beta E) / P(E)P(E) = \alpha_0 + \alpha_1 E^{1/2} + \alpha_2 E + \alpha_3 E^{1+s}, \quad (4.50)$$

where the constants α_i , β and s are tabulated in [62]. S_e is the electronic stopping, E is in MeV/amu, and S_e is in eV/10¹⁵ atoms/cm². The fit is valid in the range $0.01 \leq E \leq 100$ MeV/amu. SIMNRA uses the KKK stopping for H, D, T, ³He and ⁴He in C and Si. Nuclear stopping is calculated using the universal potential, see subsection 4.6.2.

4.6.4. SRIM stopping

SRIM is a program for calculating stopping powers of ions ions matter. SRIM stopping is obtained by calling the SRIM program, which has to be downloaded and installed separately. See the SRIM documentation for details of stopping power calculations.

4.6.5. Nuclear stopping

Nuclear stopping for all ions is calculated with the universal potential from ref. [136]. The reduced nuclear stopping s_n with the universal potential is given by

$$s_n = \frac{\ln(1 + 1.1383 \epsilon)}{2 [\epsilon + 0.01321 \epsilon^{0.21226} + 0.19593 \epsilon^{0.5}]} \quad (4.51)$$

for $\epsilon \leq 30$. For $\epsilon > 30$ s_n is given by

$$s_n = \frac{\ln(\epsilon)}{2\epsilon} \quad (4.52)$$

The reduced energy ϵ in Equation 4.51 and Equation 4.52 is calculated using the universal screening length a_U , which is $\propto 1/(Z_1^{0.23} + Z_2^{0.23})$.

4. Physics

Nuclear stopping for incident hydrogen, deuterium and tritium ions is negligible for incident energies above about 10 keV/amu [6]. Nuclear stopping for helium ions is only important at incident energies $E < 100$ keV, at higher energies nuclear stopping becomes negligible. The nuclear stopping component for heavy ions may be large and cannot be neglected.

4.6.6. Stopping in compounds

SIMNRA uses Bragg's rule [30] for the determination of the stopping power in compounds. Bragg's rule is a simple linear additivity rule of the stopping contributions of the different compound elements, assuming that the interaction of an incident ion with a target atom is independent of the surrounding target atoms. For a compound consisting of different elements i with atomic concentrations c_i ($\sum c_i = 1$) the total stopping power S is given by

$$S = \sum c_i S_i \quad (4.53)$$

S_i is the stopping power of each element.

Bragg's rule assumes that the interaction between the ion and the atom is independent of the environment. The chemical and physical state of the medium is, however, observed to have an effect on the energy loss. The deviations from Bragg's rule predictions are most pronounced around the stopping power maximum and for solid compounds such as oxides, nitrides and hydrocarbons. The deviations from Bragg's rule predictions may be of the order of 10–20% [26, 137]. For compounds with heavier atoms such as Fe_2O_3 , NbC, NbN, Ta_2O_5 , WO_3 , Au alloys etc. deviations from Bragg's rule disappear (deviation $< 2\%$) [111, 137].

Ziegler and Manoyan [137] have developed the 'cores and bonds' (CAB) model, which assumes the electronic energy loss to have two contributions: The effect of the cores and the effect of the bonds, such as C-H and C-C. The CAB-model allows better predictions for the stopping in compounds, however, the bond structure has to be known. Currently the CAB-model (or any other model which allows better predictions for the stopping in compounds) is not implemented in SIMNRA. However, a correction factor f may be used for each ion species and each layer, see section 3.6 for details. If a factor f is defined for a layer, then the program will use the stopping power $S(E)$, with E the ion energy,

$$S(E) = f S_{\text{Bragg}}(E) \quad (4.54)$$

$S_{\text{Bragg}}(E)$ is the stopping power according to Bragg's rule. Note that the factor f is energy independent.

4.7. Detector energy resolution

4.7.1. Time-of-flight detector

A time-of-flight (TOF) detector is characterized by its time resolution, which determines the energy resolution. The energy E is given by

$$\begin{aligned} E &= \frac{1}{2}mv^2 \\ &= \frac{1}{2}m\frac{L^2}{T^2}, \end{aligned}$$

where m is the ion mass, v the ion velocity, L the free flight path, and T the time of flight.

The energy resolution ΔE is determined from the time resolution ΔT according to

$$\begin{aligned} \Delta E &= m\frac{L^2}{T^3}\Delta T \\ &= 2E\frac{\Delta T}{T}. \end{aligned} \tag{4.55}$$

Equation 4.55 is used by SIMNRA to calculate the energy resolution of TOF detectors.

4.7.2. Electrostatic detector

The energy resolution of electrostatic detectors is given by the constant ratio

$$c = \frac{\Delta E}{E}, \tag{4.56}$$

with ΔE the detector energy resolution (full width at half maximum, FWHM) and E the particle energy. This relation is used for all particle species.

4.8. Straggling

4.8.1. Overview

When a beam of charged particles penetrates matter, the slowing down is accompanied by a spread in the beam energy. This phenomenon is called straggling. It is due to statistical fluctuations of the energy transfer in the collision processes.

Energy loss straggling has different contributions:

1. Electronic energy loss straggling due to statistical fluctuations in the transfer of energy to electrons.
2. Nuclear energy loss straggling due to statistical fluctuations in the nuclear energy loss.
3. Geometrical straggling due to finite detector solid angle and finite beam spot size, resulting in a distribution of scattering angles and different pathlengths for outgoing particles.
4. Straggling due to multiple small angle scattering, resulting in angular and energy spread on the ingoing and outgoing paths.
5. Straggling due to surface and interlayer roughness and thickness inhomogeneities of absorber foils.

An additional contribution to the energy broadening visible in experimental spectra is the energy resolution of the detector. The different straggling contributions (excluding roughness) have been reviewed by Szilágyi *et. al.* [107, 109], and can be included in SIMNRA calculations. The details are described in the following sections.

4.8.2. Electronic energy loss straggling

There are four main theories describing electronic energy loss straggling [24, 64, 113], each applicable in a different regime of energy loss. With ΔE the mean energy loss of the beam, and E the energy of the incident beam, we can distinguish:

$\Delta E/E < 10\%$	Vavilov's Theory [24, 118]. For thin layers and small energy losses. The energy distribution is non-Gaussian and asymmetrical. This energy range is described approximately by SIMNRA.
10 – 20%	Bohr's Theory [25, 71]. As the number of collisions becomes large, the distribution of particle energies becomes Gaussian.
20 – 50%	Symon's Theory [64]. This theory includes non-statistical broadening caused by the change in stopping power over the particle energy distribution. If the mean energy of the beam is higher than the energy of the stopping power maximum, then particles with a lower energy have a higher stopping power, and particles with higher energy have a smaller stopping power. This results in a nonstatistical broadening of the energy distribution. The width of the particles energy distribution in Symon's theory is significantly higher than predicted by Bohr's theory. The distribution of particle energies is still Gaussian.
50 – 90%	Payne's and Tschalärs Theory [91, 115, 116]. When the energy losses become very large and the mean energy of the beam decreases below the energy of the stopping power maximum, the particle energy distribution again become skewed, because now particles with lower energy have a lower stopping power than particles with higher energy. The distribution is about Gaussian.

SIMNRA 6 and earlier assumed that the particles energy distribution is Gaussian. SIMNRA 6.29 and higher takes deviations from Gaussian distributions into account by including their asymmetry via the third moment. The energy distributions are approximated with a two-piece normal distribution [78]. This improves the accuracy for very thin and very thick layers. However, for very thin layers this is still an approximation: In this case the energy distribution is described by the Vavilov distribution [24, 118]. However, the straggling contribution of thin layers to the total energy broadening is much smaller than the contribution of the finite energy resolution of the detector, so that for typical IBA applications the total energy broadening is still well described even for very thin layers.

SIMNRA calculates the non-statistic broadening or skewing (often called propagation of straggling [109]) of the energy distribution when penetrating a layer in the following way:

Assume two particles with energies E_1 and E_2

$$\begin{aligned} E_1 &= E_0 + \frac{\Delta E}{2} \\ E_2 &= E_0 - \frac{\Delta E}{2} \end{aligned}$$

centered around a mean energy E_0 . The energy difference $E_1 - E_2$ of the two particles is ΔE . After penetrating a layer the particles have the energies E'_1 and E'_2 centered around a mean

4. Physics

energy E'_0 . The energy difference $\Delta E'$ behind the layer is given by

$$\Delta E' = \frac{\epsilon(E'_0)}{\epsilon(E_0)} \Delta E = \frac{\epsilon_f}{\epsilon_i} \Delta E \quad (4.57)$$

with the stopping power $\epsilon = dE/dx$. ϵ_f is the stopping power at the exit of the layer and ϵ_i the stopping power at the entrance of the layer. If $\epsilon_f > \epsilon_i$, which is the case for all energies above the stopping power maximum, the energy difference increases and the distribution function is broadened. If $\epsilon_f < \epsilon_i$, the energy difference decreases and the distribution function gets skewed.

To the non-statistic broadening we have to add the statistical effects. When the incident beam with initial energy E_0 and initial beam width σ_0^2 (σ^2 is the variance of the energy distribution, the full width at half maximum (FWHM) is $2\sqrt{2\ln 2}\sigma = 2.355\sigma$) penetrates a layer of matter with thickness Δx , then the beam width σ_1^2 after penetrating the layer is given by:

$$\sigma_1^2 = \left(\frac{\epsilon_f}{\epsilon_i}\right)^2 \sigma_0^2 + \sigma^2 \quad (4.58)$$

ϵ_f and ϵ_i are the stopping powers of the material at the entrance and exit of the layer and σ^2 is the energy loss straggling in the layer. The first term in Equation 4.58 describes the non-statistical broadening of the beam according to Equation 4.57 due to the energy dependence of the stopping power, the second term adds the statistical effects.

The electronic energy-loss straggling in Bohr approximation σ_{Bohr}^2 is given by [25, 71]:

$$\sigma_{Bohr}^2 [\text{keV}^2] = 0.26 Z_1^2 Z_2 \Delta x [10^{18} \text{ atoms/cm}^2] \quad (4.59)$$

In Chu's theory the Bohr straggling is modified by a correction factor H [35, 71]:

$$\sigma_{Chu}^2 = H(E/M_1, Z_2) \sigma_{Bohr}^2 \quad (4.60)$$

Bohr's theory of electronic energy loss straggling is valid in the limit of high ion velocities. In this case the electronic energy loss straggling is almost independent of the ion energy. For lower ion energies the Bohr straggling is multiplied by the Chu correction factor $H(E/M_1, Z_2)$, which depends only on E/M_1 and the nuclear charge of the target atoms Z_2 . H takes into account the deviations from Bohr straggling caused by the electron binding in the target atoms. Chu [35, 71] has calculated H by using the Hartree-Fock-Slater charge distribution. This calculation gives straggling values which are considerably lower than those given by Bohr's theory. The correction factor H , as used by SIMNRA if Chu straggling is selected, is shown in Figure 4.8. The Z_2 oscillations are clearly visible. The Chu correction is mainly necessary for high Z_2 and low energies. For high energies H approaches 1 and becomes independent of Z_2 and energy. For E/M_1 values in the range 100–1000 keV/amu the data have been taken from ref. [71], data for lower and higher E/M_1 values are based on an extrapolation performed in ref. [109]. Tabulated values for H are stored in the file CHU_CORR.DAT. For not tabulated values SIMNRA uses linear interpolation.

Charge state fluctuations of the ions result in an additional straggling contribution, which was taken into account empirically by Yang et al. [128]. In Yang's theory, the total straggling is expressed by

$$\frac{\sigma_{Yang}^2}{\sigma_{Bohr}^2} = \gamma^2(Z_1, Z_2, v) \frac{\sigma_{Chu}^2}{\sigma_{Bohr}^2} + \frac{\Delta\sigma^2}{\sigma_{Bohr}^2}, \quad (4.61)$$

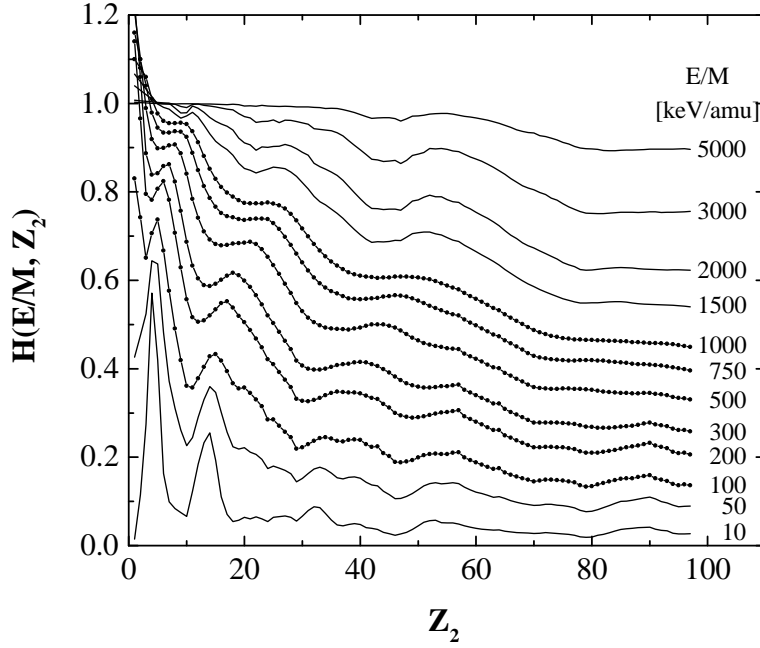


Figure 4.8.: The Chu straggling correction for several values of E/M_1 as a function of the nuclear charge of the target Z_2 . Dots are original data from Chu [71], solid lines are extrapolated data taken from [109].

where $\gamma(Z_1, Z_2, v)$ is the effective charge factor for ions in matter, v the ion velocity, and $\Delta\sigma^2$ is the additional straggling due to correlation effects.

For hydrogen ions in matter, Yang always assumes a charge state of one, i.e. $\gamma = 1$. A fit to about 600 different empirical data points with a deformed resonant function gave

$$\left(\frac{\Delta\sigma^2}{\sigma_{Bohr}^2} \right)_H = \frac{B_1\Gamma}{(E - B_2)^2 + \Gamma^2}, \quad (4.62)$$

with

$$\Gamma = B_3[1 - \exp(-B_4E)], \quad (4.63)$$

where E is the energy in MeV/amu and B_1 – B_4 are fitted constants.

Helium and heavier ions are not always fully stripped, but the charge state is a constantly changing parameter, fluctuating along the trajectory. The effective charge state $\gamma(Z_1, Z_2, v)$ was calculated by Ziegler [136], and is also used by Yang. The extra contribution to energy straggling $\Delta\sigma^2$ of helium and heavy ions is fitted by

$$\left(\frac{\Delta\sigma^2}{\sigma_{Bohr}^2} \right)_{HI} = \frac{Z_1^{4/3}}{Z_2^{1/3}} \frac{C_1\Gamma}{(\mathcal{E} - C_2)^2 + \Gamma^2}, \quad (4.64)$$

with

$$\Gamma = C_3[1 - \exp(-C_4\mathcal{E})], \quad (4.65)$$

4. Physics

and

$$\mathcal{E} = \frac{E}{Z_1^{3/2} Z_2^{1/2}}, \quad (4.66)$$

where E and \mathcal{E} are the energies in MeV/amu and C_1 – C_4 are fitted constants. The above equations are for solid targets.

SIMNRA 6.0x used the fit from [109] for the prefactor H from Chu's theory and a slightly simplified expression for the effective charge factor γ . The fit for the prefactor H from Chu's theory by Yang and the full expression for the effective charge factor γ were implemented in SIMNRA 6.98. The specific straggling as calculated by Yang in [128] is compared to SIMNRA 6.06 and SIMNRA 6.98 in Figure 4.9. SIMNRA 6.98 is in excellent agreement with Yang for protons, helium, and heavy ions. SIMNRA 6.06 is in very good agreement with the original Yang data for heavy ions, but minor differences $\leq 10\%$ are observed for protons and helium ions.

Figure 4.10 compares the beam widths (FWHM) of 2.5 MeV ^4He ions in silicon calculated by SIMNRA using different straggling models with Bohr's theory. The SIMNRA calculation always includes propagation of straggling according to Equation 4.57. For small energy losses the beam width calculated by SIMNRA using Bohr straggling is identical to the simple Bohr theory from Equation 4.59. At larger energy losses the SIMNRA beam width exceeds Bohr straggling due to the nonstochastic broadening. With Chu straggling the width is somewhat smaller than Bohr due to the binding correction, while charge state fluctuations result in an additional broadening within the Yang theory. When the mean beam energy has decreased below the energy of the stopping power maximum, the beam width becomes skewed.

As can be seen from Figure 4.8 the deviation of the Chu correction from Bohr's theory is largest for high Z_2 and low energies. Figure 4.11 shows the beam width (FWHM) of 1 MeV ^4He penetrating through gold. The stopping power maximum is at about 960 keV, and non-stochastic skewing plays a role. Therefore even with Bohr straggling the deviation from Bohr's theory gets large at larger energy losses due to the skewing effect. Due to the large Chu correction even at small energy losses the deviation from the simple Bohr theory is large.

Figure 4.12 shows the measured RBS-spectrum for 1.0 MeV ^4He ions incident on a gold layer with a thickness of about 100 nm and a scattering angle of 165° compared with simulations using Bohr straggling and Chu straggling. As can be seen at the low energy edge of the layer, the Bohr straggling is broader than the experimental data. The Chu straggling fits the measured curve relatively well, except of the multiple scattering contribution.

The straggling of outgoing particles is calculated with Equation 4.58 as well. Outgoing particles always start with an energy distribution with variance σ_{out}^2 which is given by

$$\sigma_{\text{out}}^2 = K^2 \sigma_{\text{in}}^2 \quad (4.67)$$

with K the kinematic factor and σ_{in}^2 the variance of the energy distribution of the incident beam.

Accuracy of electronic energy-loss straggling calculations

SIMNRA uses Equation 4.58 for the calculation of straggling. However, Equation 4.58 is only correct for infinitesimal thin layers, i.e. if Δx (see Equation 4.59) is sufficiently small. It gets more and more inaccurate for thicker layers, because the straggling propagation for the second term in Equation 4.58 is neglected.

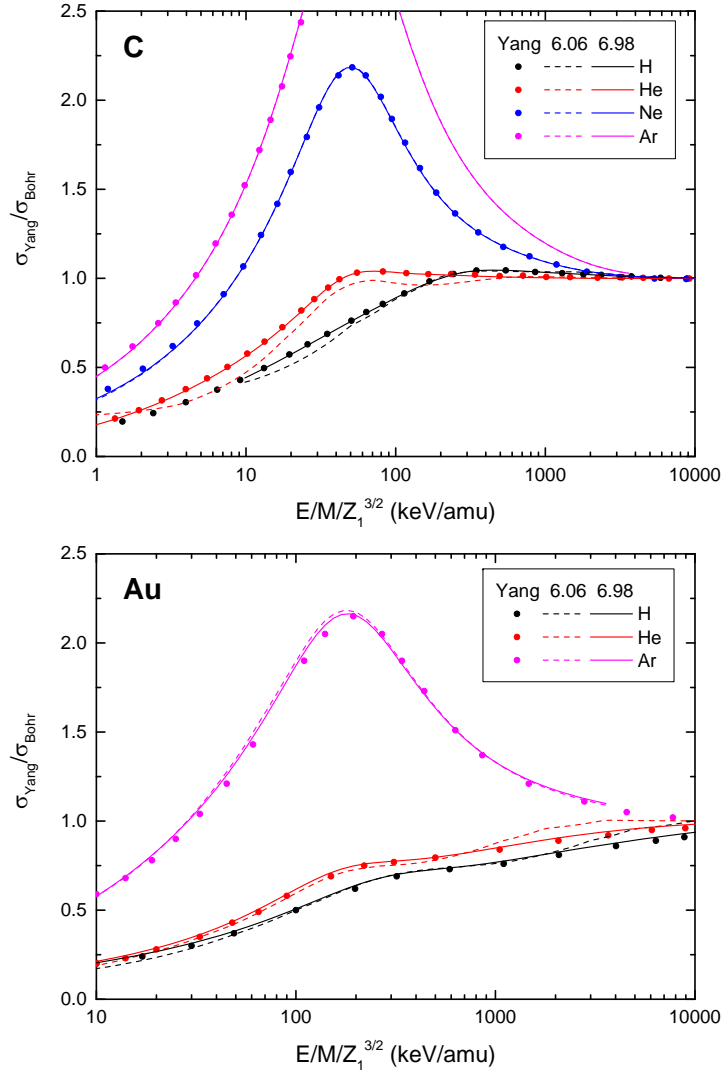


Figure 4.9.: Comparison of the specific Yang straggling $\sigma_{\text{Yang}}/\sigma_{\text{Bohr}}$ for various projectiles in carbon (top) and gold (bottom). Dots: Original data by Yang from [128]; Dashed lines: SIMNRA 6.06; Solid lines: SIMNRA 6.98 and later.

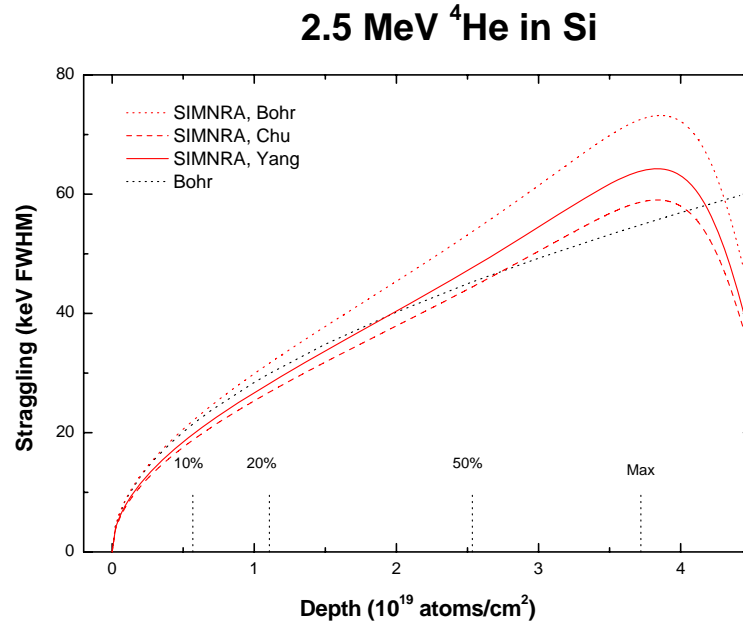


Figure 4.10.: Beam width (FWHM) of 2.5 MeV ^4He ions penetrating through silicon. Black dotted line: Bohr's theory from Equation 4.59; red dotted line: Bohr's theory including propagation of straggling, as calculated by SIMNRA; red dashed line: Chu's theory including propagation; red solid line: Yang's theory including propagation. The vertical lines denote the mean depth at which the beam has lost 10%, 20% and 50% of its initial energy. Max denotes the depth at which the mean energy of the beam has decreased to the energy of the stopping power maximum.

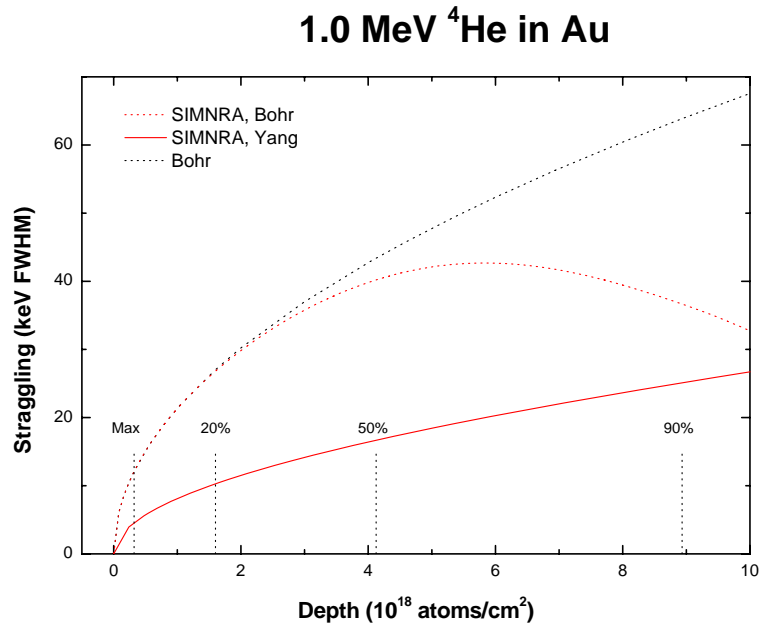


Figure 4.11.: Beam width (FWHM) of 1.0 MeV ^4He ions penetrating through gold. Black dotted line: Bohr's theory from Equation 4.59; red dotted line: Bohr's theory including propagation of straggling, as calculated by SIMNRA; red solid line: Yang's theory including propagation. The vertical lines denote the mean depth at which the beam has lost 20%, 50% and 90% of its initial energy. Max denotes the depth at which the mean energy of the beam has decreased to the energy of the stopping power maximum.

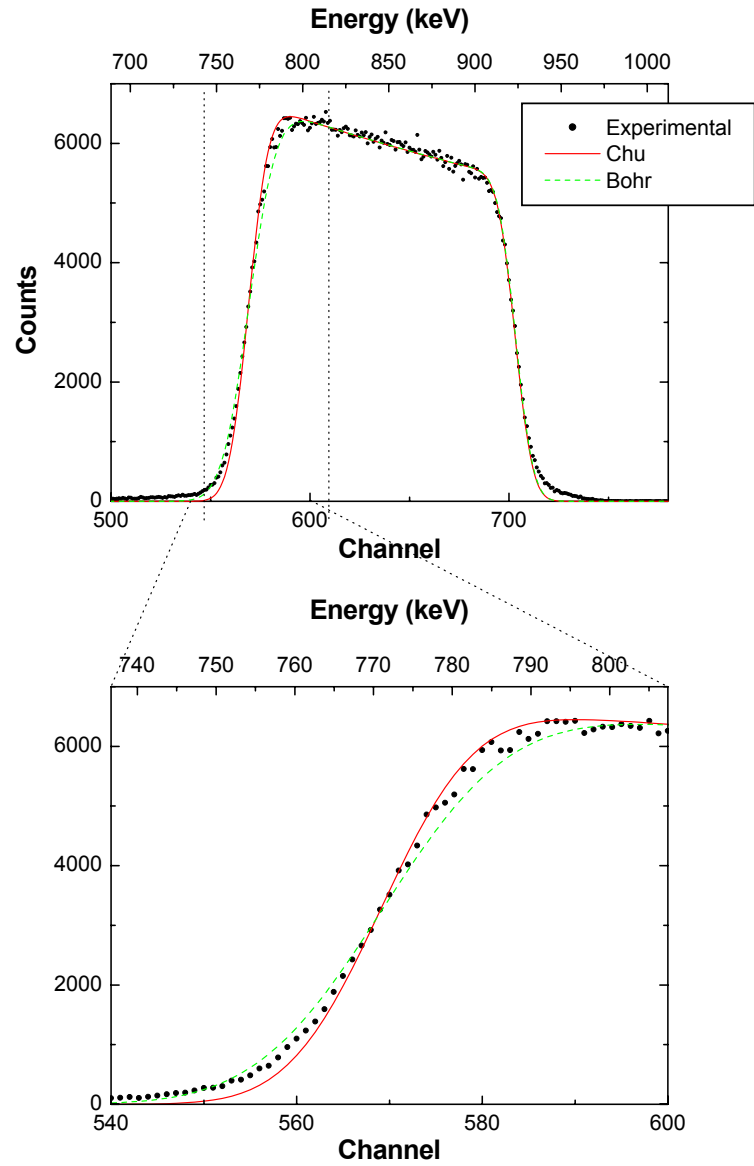


Figure 4.12.: Measured and simulated spectra using Bohr and Chu straggling of 1.0 MeV ^4He ions incident on 100 nm Au on Si, scattering angle 165° . Bottom: Magnification of the low energy gold edge.

The error $\Delta\sigma^2$ for σ_1^2 (see Equation 4.58) can be estimated to be

$$\Delta\sigma^2 \leq \begin{cases} \left| 1 - \left(\frac{\epsilon_f}{\epsilon_i} \right)^2 \right| G\Delta x & \text{if } |1 - (\epsilon_f/\epsilon_i)^2| > 0.03 \\ 0.03 G\Delta x & \text{else.} \end{cases} \quad (4.68)$$

G depends on theory: For Bohr straggling it is given by Equation 4.59, and for Chu and Yang straggling by Equation 4.60 and Equation 4.61, respectively. 0.03 is based on extensive tests of Equation 4.68: Using only the upper half of the equation results in unrealistically small error estimates (and too large step widths) around the stopping power maximum, where $\epsilon_f \approx \epsilon_i$ on both sides of the maximum.

The step width Δx for electronic energy-loss straggling calculations is selected in such a way, that

$$\frac{\Delta\sigma^2}{\sigma_1^2} \leq \delta, \quad (4.69)$$

where δ is the desired accuracy. SIMNRA uses a default of $\delta = 1\%$. An estimate of the accuracy of straggling calculations can be obtained with the program VIEWNRA.

An example for the accuracy of straggling calculations and the error estimate is shown in Figure 4.13. The error of the straggling calculation is obtained by comparison to a calculation with very small fixed step width. The error estimate based on Equation 4.68 gives the correct sign and magnitude of the real error.

4.8.3. Nuclear energy loss straggling

Fluctuations in the number of nuclear collisions lead to nuclear energy loss straggling, which can be described by Bohr's theory of nuclear straggling. The variance σ_n^2 in Bohr's approximation is given by:

$$\sigma_n^2 [\text{keV}^2] = 0.26 Z_1^2 Z_2^2 \left(\frac{M_1}{M_1 + M_2} \right)^2 \Delta x [10^{18} \text{ atoms/cm}^2] \quad (4.70)$$

Nuclear energy loss straggling is small compared to electronic energy-loss straggling for light ions (protons or helium ions), and can be neglected. Nuclear straggling becomes more important for heavy ions, and the variance of nuclear straggling may exceed electronic energy-loss straggling, see Equation 4.59 and Equation 4.70. However, it has been shown experimentally for 60 MeV ^{58}Ni ions in different target materials, that the width of the total straggling distribution is only somewhat larger than electronic energy loss straggling alone [97]. The nuclear straggling energy distribution is broad with a long tail towards low energies, which cannot be described by a Gaussian distribution. This tail leads to the large variance of nuclear straggling. However, the tail contains only a small fraction of all particles, and the width of the total energy distribution (electronic plus nuclear straggling) is still dominated by electronic energy-loss straggling [97]. This justifies to neglect nuclear straggling for all ion species.

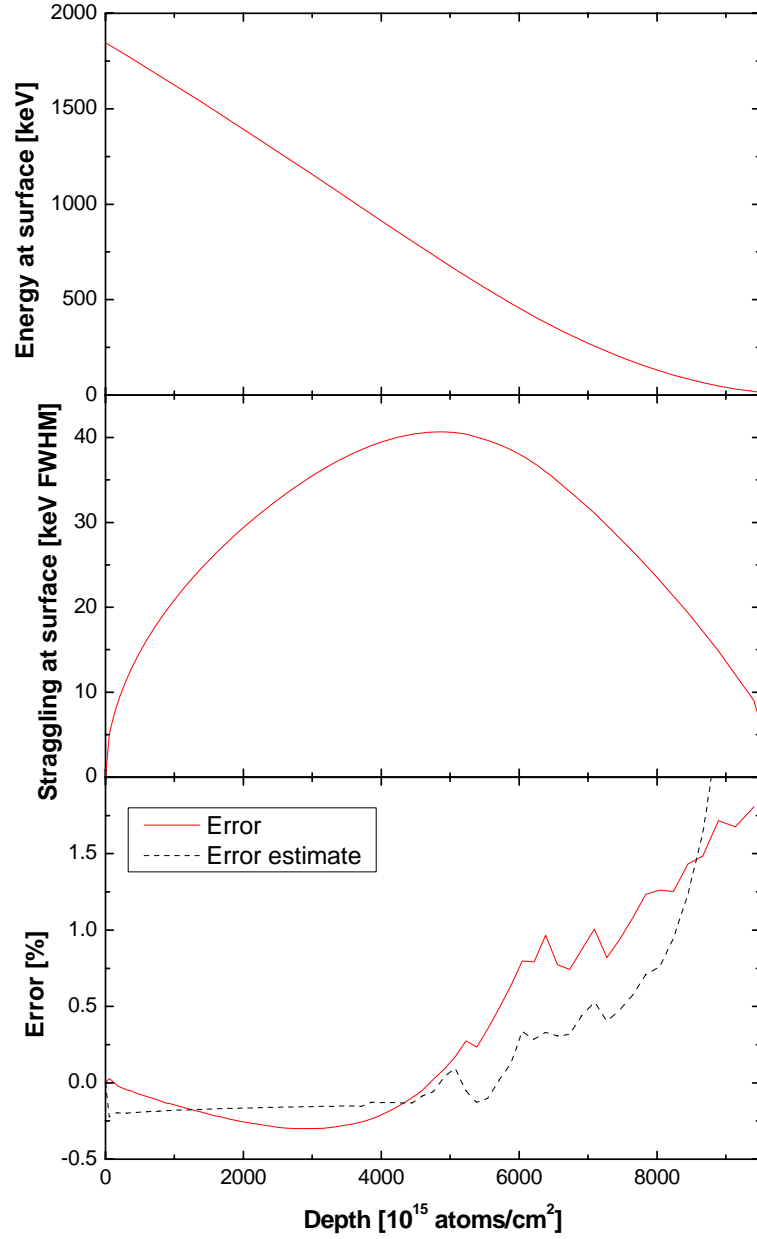


Figure 4.13.: Energy and energy loss straggling of backscattered ^4He ions at the target surface in Au. Incident angle $\alpha = 0^\circ$, exit angle $\beta = 15^\circ$, scattering angle $\theta = 165^\circ$, Yang straggling. Top: Energy of backscattered ions at the target surface; Middle: Electronic energy-loss straggling at the target surface; Bottom: Error of the straggling calculation and error estimate based on Equation 4.68.

4.8.4. Energy loss straggling in compounds

For compounds a simple additivity rule for energy loss straggling is used [71]. The straggling in a compound consisting of elements i with atomic concentration c_i is calculated with

$$\sigma^2 = \sum_i c_i \sigma_i^2 \quad (4.71)$$

with σ_i^2 being the straggling in each element.

4.8.5. Geometrical straggling

The finite size of the incident beam and the width of the detector aperture result in a spread $\Delta\beta$ of the exit angle β for outgoing particles. This angular spread leads to different energies of the particles at the target surface due to

1. a spread $\Delta\theta$ of the scattering angle θ and therefore a spread of the transferred energy in the scattering process
2. different path lengths of the outgoing particles in the material.

These two contributions to geometrical straggling are not independent of each other and have to be considered simultaneously.

The spread $\Delta\beta$ of the exit angle β is given by [38, 109]:

$$(\Delta\beta)^2 = \left(\frac{g_d w}{L_D} \right)^2 + \left(\frac{g_b d \cos \beta}{L_D \cos \alpha} \right)^2 \quad (4.72)$$

where d is the diameter of the incident beam⁵, w is the width of the detector aperture and L_D is the distance between the sample and the detector aperture, see Figure 3.9. g_b and g_d take into account the shapes of the beam and the detector aperture. For circular detectors or circular beams with uniform current density $g = 0.59$. $g = 0.68$ for rectangular shapes, as in the case of narrow slits [38].

SIMNRA calculates geometrical straggling by calculating the energy at the surface of particles with exit angles $\beta - \Delta\beta/2$ and $\beta + \Delta\beta/2$ and corresponding scattering angles $\theta - \Delta\theta/2$ and $\theta + \Delta\theta/2$. The spread of θ is given by

$$\Delta\theta = \frac{\partial\theta}{\partial\beta} \Delta\beta. \quad (4.73)$$

This equation can be solved only if the relation between β and θ is known.

In IBM geometry (see Figure 3.17 and section 3.8) the incident and outgoing trajectories are in the same plane with

$$180^\circ = \alpha + \beta + \theta. \quad (4.74)$$

The spread in θ is then simply given by

$$\Delta\theta = -\Delta\beta. \quad (4.75)$$

⁵Note that the beam spot size on the sample surface is $d/\cos \alpha$.

4. Physics

Note the minus sign: If β increases then θ decreases and vice versa.

In Cornell geometry (see [Figure 3.17](#) and [section 3.8](#)) the relation between the angles α , β and θ is given by

$$\cos \beta = -\cos \theta \cos \alpha, \quad (4.76)$$

and the spread of θ can be obtained from

$$\frac{\partial \theta}{\partial \beta} = -\frac{1}{\sin \theta} \frac{\sin \beta}{\cos \alpha}. \quad (4.77)$$

In general geometry the relation between the angles α , β and θ is undefined, and $\partial \theta / \partial \beta$ cannot be determined. *Calculation of geometrical straggling is not possible for general geometry.* Geometrical straggling is only calculated for IBM and Cornell geometries. SIMNRA will automatically determine if IBM or Cornell geometry is present.

Energy loss straggling, geometrical straggling and the detector resolution are independent and are added quadratically.

[Figure 4.14](#) gives as examples the different straggling contributions at the sample surface for 2.6 MeV ^4He incident on $\text{C}_{0.99}\text{D}_{0.01}$ as a function of depth for typical RBS and ERDA geometries. An energy independent detector resolution of 15 keV FWHM is used. For RBS geometry the contribution of geometrical straggling is small compared to energy loss straggling and the resolution of the detector and may be neglected without penalty. For the ERDA geometry however, geometrical straggling cannot be neglected. Note that geometrical straggling first decreases until it reaches zero and then increases again. This is due to the minus-sign in [Equation 4.75](#): If the exit angle β decreases (the outgoing path is closer to the surface normal) then the scattering angle θ increases: The backscattered or recoiled particles start with a smaller energy. However, the path length in the material is smaller due to the exit angle which is closer to normal resulting in a smaller energy loss. Near the surface the path length differences are small and geometrical straggling is governed by the kinematic spread. With increasing depth the two effects compensate each other more and more until in large depths the path length differences become dominant and geometrical straggling increases.

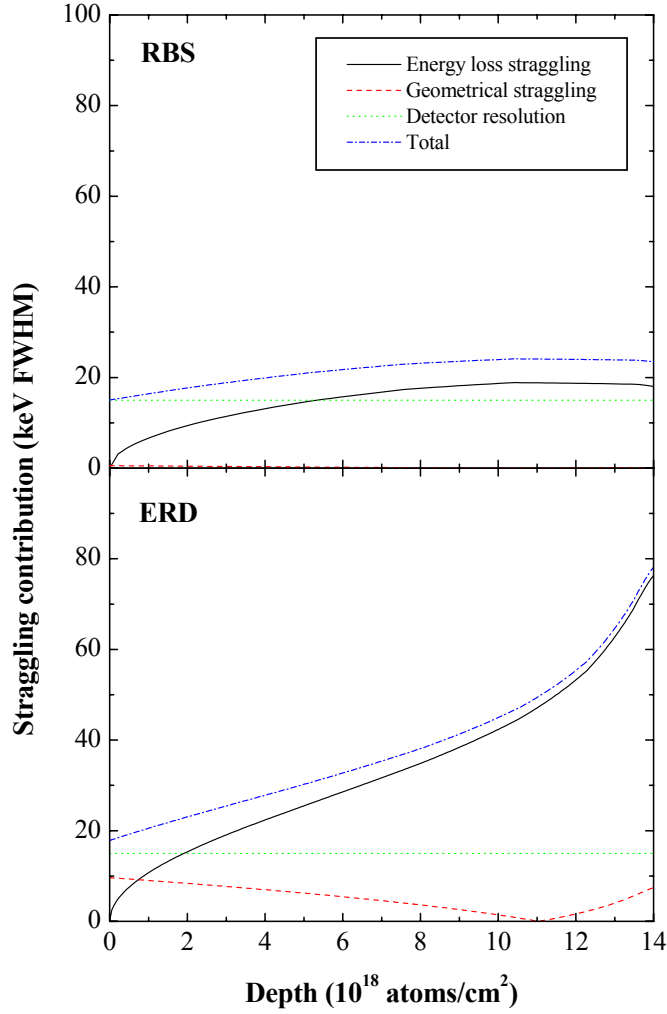


Figure 4.14.: Contributions of electronic energy loss straggling, geometrical straggling and detector resolution to the total energy straggling at the sample surface for 2.6 MeV ^4He incident on $\text{C}_{0.99}\text{D}_{0.01}$. Beam diameter 0.5 mm, aperture width 0.5 mm, distance sample-detector aperture 100 mm, detector resolution 15 keV FWHM.

Top: RBS geometry with $\theta = 165^\circ$, $\alpha = 0^\circ$, $\beta = 15^\circ$ for ^4He backscattered from ^{12}C .

Bottom: ERDA geometry with $\theta = 30^\circ$, $\alpha = 75^\circ$, $\beta = 75^\circ$ for the deuterium recoils.

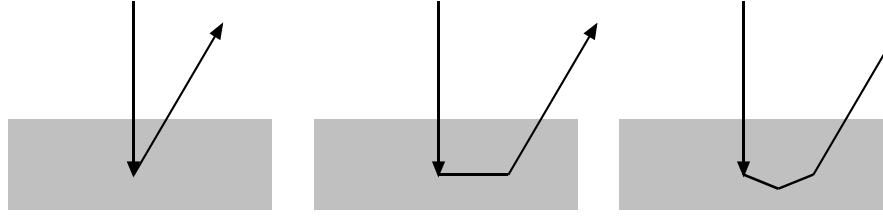


Figure 4.15.: Examples of ion trajectories with one, two and three scattering events.

4.9. Multiple and plural scattering

4.9.1. Overview

SIMNRA uses straight lines as trajectories for the ingoing and outgoing particles, with one single scattering event connecting the trajectories of the particles, see Figure 4.15 left. This is only an approximation to physical reality, because the particles on the ingoing and outgoing path suffer many small angle deflections with small scattering angles (this has been called multiple scattering) and additionally may perform more than one scattering event with large scattering angle (see Figure 4.15 middle and right), before they are scattered towards the detector. This has been called plural scattering.

Multiple scattering has been reviewed by Szilágyi *et al.* [109] and Amsel *et al.* [2]. Multiple scattering results in an angular spread of the particles and therefore in a spread of path lengths. Due to the path length differences, we get an energy spread of the particles in a given depth.

Plural scattering with 2, 3, 4, ... scattering events is responsible for the background behind the low energy edge of high Z elements on top of low Z elements and the steeper increase of the spectra towards low energies than calculated with single scattering [19, 42, 104, 123, 124]. SIMNRA is able to calculate dual scattering, i. e. all particle trajectories with two scattering events.

4.9.2. Multiple (small angle) scattering

Angular and lateral spread due to multiple collisions with small deflection angles was treated analytically and numerically by Sigmund *et al.* [70, 102] for realistic screened interaction potentials of the Thomas-Fermi and Lenz-Jensen type. However, their theory is only valid for

1. Single elemental targets
2. The stopping of ions is neglected.

Due to these two assumptions the original Sigmund theory is only of limited practical use. Szilágyi *et al.* [109] and Amsel *et al.* [2] have proposed a method how to extend the original Sigmund theory to multi-elemental targets with stopping, thus yielding the angular spread distribution for realistic cases. They also provide fit functions for the universal Sigmund function of (reduced) angular spread as function of (reduced) target depth. These fit functions are slightly different and can be selected in the *Calculation:Parameter* tab, see subsection 3.5.3. A comparison

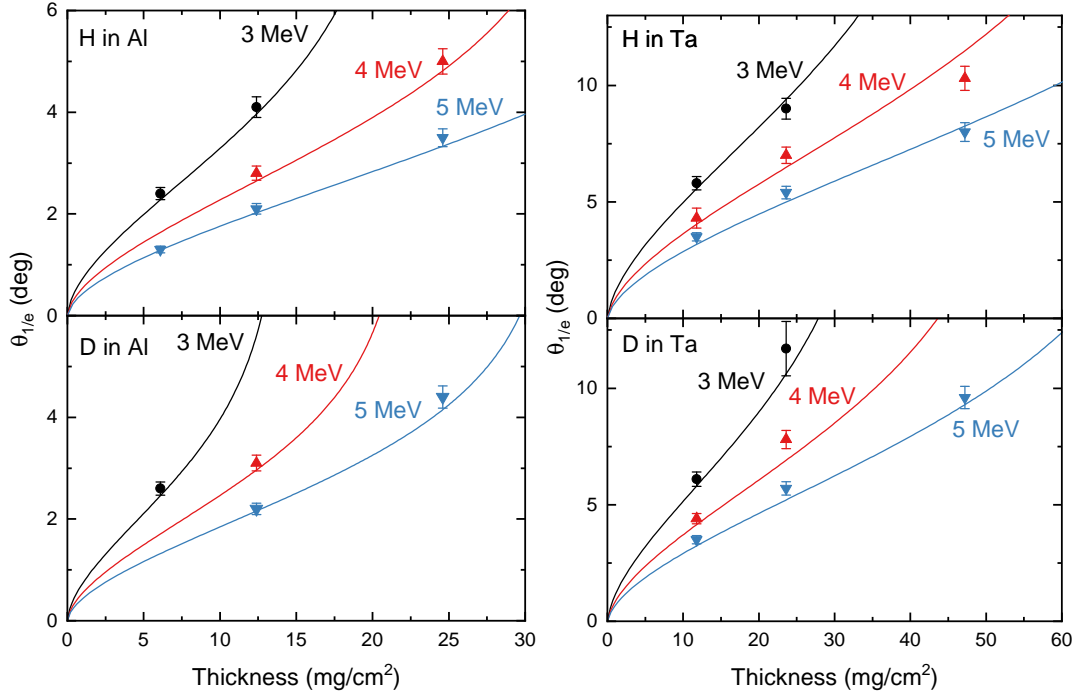


Figure 4.16.: Width of the lateral spread distribution by multiple-scattering of 3–5 MeV protons (top) and deuterons (bottom) in aluminum (left) and tantalum (right) as function of depth. The width is given at $\theta_{1/e}$, i.e. the angle at which the distribution decreased to 1/e of its maximum value. Symbols: Experimental data from [39]; Solid lines: SIMNRA.

of the angular spread by multiple scattering of 3–5 MeV protons and deuterons in Al and Ta as calculated by SIMNRA with experimental data from [39] is shown in Fig. 4.16. SIMNRA agrees with the experimental data typically within 5–10%.

Szilagyi *et al.* [109] and Amsel *et al.* [2] also proposed an approximation how to obtain the energy spread distribution from the angular spread distribution via the chord angle. Angular spread and energy spread are strongly correlated [2, Section 4]. This correlation is taken into account in the calculation of the energy of backscattered or recoiled particles.

The Szilagyi/Amsel theory was initially implemented in the DEPTH code [108, 109], which is primarily intended for depth-resolution calculations. SIMNRA contains an independent implementation of the Szilagyi/Amsel theory⁶. A comparison of DEPTH, SIMNRA, Monte-Carlo calculations in binary collision approximation, and molecular dynamics calculations, can be found in [77].

When using multiple scattering calculations, it should be kept in mind that SIMNRA uses the following approximations:

1. Multiple scattering energy spread distributions are approximated by Gaussian functions.

⁶The full Szilagyi/Amsel theory is implemented in SIMNRA 5.76 and higher. Earlier versions of SIMNRA treated the correlation between angular and energy spread only approximately, thus resulting in somewhat different results.

4. Physics

In reality, these distributions have stronger wings, which are underestimated by SIMNRA.

2. Energy transfer to target atoms (i.e. nuclear stopping and straggling) results in low-energy tails. These tails are neglected.

4.9.3. Plural (large angle) scattering

Plural large angle scattering cannot be treated analytically. SIMNRA approximates plural scattering effects by calculating all particle trajectories with two scattering events (dual scattering), see Figure 4.15. The contribution of trajectories with more than two large angle deflections is neglected. Plural (large angle) and multiple (small angle) scattering may be combined in the calculations.

SIMNRA performs the calculation of dual scattering in the following way: During each step of the incident ion particles are scattered into the whole sphere of 4π . We introduce the polar system with the polar angles ψ , φ , see Figure 4.17: ψ is the altitude, and φ the azimuth angle. After the first scattering event the scattered particles have the direction ψ , φ . The scattering angle θ_1 of the first scattering event is given by

$$\cos \theta_1 = \sin \alpha \sin \psi \sin \varphi - \cos \alpha \cos \psi$$

SIMNRA uses the Rutherford cross-section for the calculation of the number of scattered particles in the first scattering event. The new angle α' of the particles after the first scattering is

$$\alpha' = 180^\circ - \psi$$

The scattering angle θ_2 of the second scattering event is given by⁷

$$\cos \theta_2 = \sin \beta \cos \varphi_\beta \sin \psi \cos \varphi + \sin \beta \sin \varphi_\beta \sin \psi \sin \varphi + \cos \beta \cos \psi, \quad (4.78)$$

with φ_β the azimuth angle of the exit vector $\vec{\beta}$. SIMNRA subdivides the whole sphere of 4π into 10 ψ -intervals and 12 φ -intervals, resulting in 120 solid angle intervals. SIMNRA considers only trajectories with scattering angles $\theta_1, \theta_2 > 20^\circ$ for dual scattering. Trajectories with smaller scattering angles are very similar to single scattering trajectories. For each solid angle interval a full backscattering spectrum with the starting depth of the particles equal to the depth of the incident ions and the new incident angle α' and the new scattering angle θ_2 is calculated.

Figure 4.18 compares the simulated spectra with single and dual scattering for 500 keV ^4He ions incident on a 100 nm gold layer on top of silicon with experimental data. With the inclusion of dual scattering the experimental results are much better approximated. Dual scattering gives the background between the low energy edge of Au and the Si edge, and the steeper increase of the gold spectrum is better described. The results with dual scattering are slightly lower than the experimental results. This is due to trajectories with more than two scattering events which are not calculated.

⁷Equation 4.78 is valid in general geometry. SIMNRA version 6.03 and earlier used an equation which was only valid in IBM geometry.

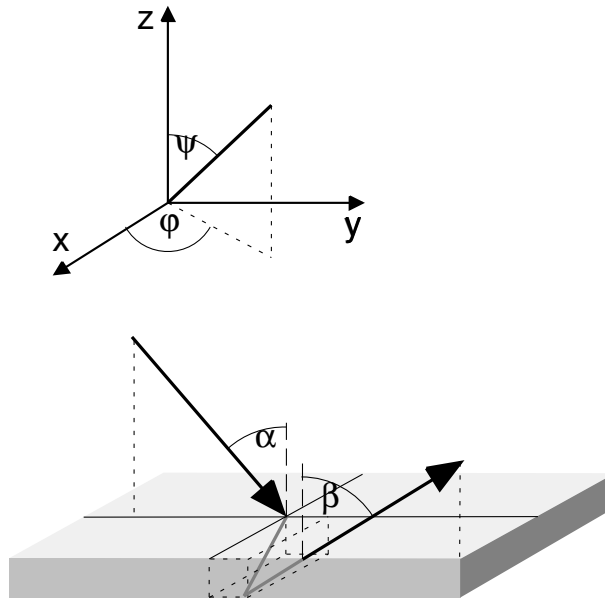


Figure 4.17.: Geometry used for the calculation of dual scattering, with the altitude angle ψ and azimuth angle φ .

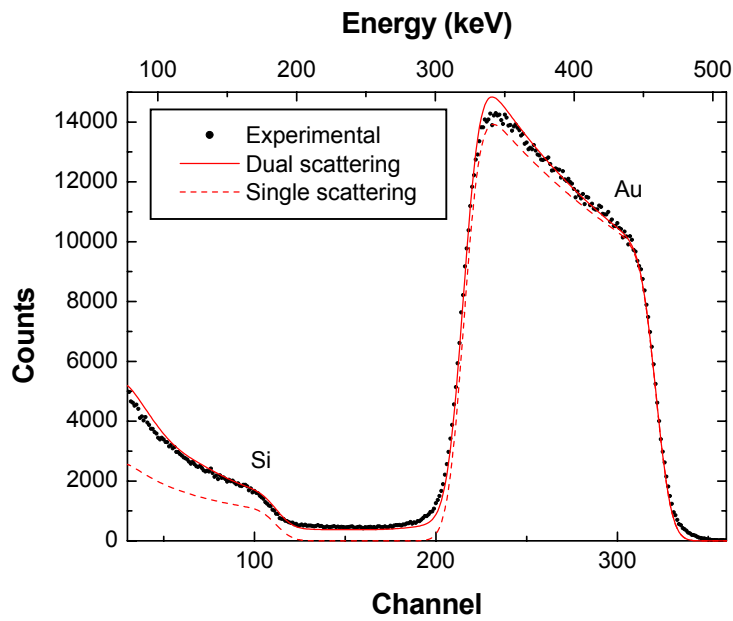


Figure 4.18.: 500 keV ^4He ions incident on 100 nm Au on top of Si, scattering angle 165° . Circles: experimental data points, dashed line: simulation with one scattering event, solid line: simulation with two scattering events.

4.10. Skewness of energy distributions

As already described in subsection 4.8.2 and section 4.9 the slowing down of swift ions in matter is always associated with energy spread due to energy-loss straggling [99] and multiple small-angle scattering [2, 107, 109]. Ion-beam analysis simulation codes (including SIMNRA 6.30 and earlier) approximate these energy spread distributions by Gaussian functions. However, as has already been shown experimentally for energy-loss straggling [115, 116] and by Monte-Carlo simulations for multiple small-angle scattering [77], these energy spread distributions are usually of non-Gaussian shape and skewed.

An accurate calculation of the shape of electronic energy-loss straggling distributions is possible within the framework of a general stopping and straggling theory, such as the Bohr, Bethe, or Bloch theory [99, 115, 116]. This requires an additional numerical convolution, which increases the computing time by about one order of magnitude: This is impractical due to computing time restrictions in simulation codes, which are often used interactively. Another limitation is, that the shape of the straggling distribution cannot be calculated if empirical or semi-empirical stopping or straggling models (such as SRIM stopping powers [135] or Yang straggling [128]) are used, because these models provide only information about one specific moment of the energy distribution: A stopping power model gives the mean energy loss and mean energy, a straggling model the variance of the straggling. But it is usually impossible to get information about higher moments (such as the skewness and kurtosis) from such semi-empirical models. Because these models often provide a better accuracy than a general stopping and straggling theory, such semi-empirical models are often used in practice.

The available analytical theory of multiple small-angle scattering [2, 107, 109] intrinsically assumes symmetry of the angular and energy spread distributions and does not provide information about their skew. Monte-Carlo simulation codes, such as MCERD [9] and CORTEO [93], allow a precise calculation of multiple scattering energy distributions, but these codes are slower by three to four orders of magnitude than the above mentioned single scattering codes.

The different energy spread distributions relevant for IBA (electronic energy-loss straggling, geometrical straggling, multiple scattering, energy spread due to porosity) can be approximated with sufficient accuracy by two-piece normal distributions (TPNDs). This slows down calculations only marginally and provides energy distributions with correct mean value, variance and third moment, thus resulting in higher accuracy in spectrum simulation. From the different classes of skewed Gaussian-type functions (Gram-Charlier expansion with linear term [46, section 3.15.8], skew-normal distribution [11], and other), the TPND has been selected due to its good compatibility with the calculational schemes for IBA.

4.10.1. The two-piece normal distribution (TPND)

The two-piece normal distribution (TPND) (sometimes called binormal distribution or joined half-Gaussian) $f(x)$ is defined by

$$f(x) = \begin{cases} A \exp\left(-\frac{(x-x_0)^2}{2\sigma_1^2}\right) & \text{if } x \leq x_0 \\ A \exp\left(-\frac{(x-x_0)^2}{2\sigma_2^2}\right) & \text{if } x > x_0 \end{cases} \quad (4.79)$$

4.10. Skewness of energy distributions

x_0 is the mode of the distribution, σ_1 and σ_2 are the standard deviations towards lower and higher values. A is a normalization factor. The half widths at half maximum (HWHMs) towards lower and higher values, h_1 and h_2 , are connected to the corresponding standard deviations through

$$h_i = \sqrt{2 \ln 2} \sigma_i, \quad (4.80)$$

where i is either 1 or 2.

The moments of a TPND can be found for example in [47]. The mean value $F = \int f(x) x dx$ is given by

$$F = x_0 + \sqrt{\frac{2}{\pi}} (\sigma_2 - \sigma_1). \quad (4.81)$$

The variance $V = \int f(x) (x - F)^2 dx$ is

$$V = \sigma_1 \sigma_2 + \left(1 - \frac{2}{\pi}\right) (\sigma_2 - \sigma_1)^2 \quad (4.82)$$

The third central moment $M = \int f(x) (x - F)^3 dx$ is

$$M = \sqrt{\frac{2}{\pi}} \sigma_1 \sigma_2 (\sigma_2 - \sigma_1) + \sqrt{\frac{2}{\pi}} \left(\frac{4}{\pi} - 1\right) (\sigma_2 - \sigma_1)^3 \quad (4.83)$$

$$\approx \sqrt{\frac{2}{\pi}} \sigma_1 \sigma_2 (\sigma_2 - \sigma_1) \quad \text{for } \sigma_1 \approx \sigma_2. \quad (4.84)$$

For not too strongly skewed distributions, i.e. if the difference between σ_1 and σ_2 is below about 30%, x_0 , σ_1 and σ_2 can be calculated analytically for given F , V and M :

$$\sigma_1 = -\frac{\beta}{2} + \sqrt{\frac{\beta^2}{4} - a\beta^2 + V} \quad (4.85)$$

$$\sigma_2 = \frac{\beta}{2} + \sqrt{\frac{\beta^2}{4} - a\beta^2 + V} \quad (4.86)$$

$$x_0 = F - \frac{M}{V} \quad (4.87)$$

with the abbreviations

$$a = 1 - \frac{2}{\pi} \approx 0.3634 \quad (4.88)$$

and

$$\beta = \sqrt{\frac{\pi}{2} \frac{M}{V}}. \quad (4.89)$$

Equation 4.85 and Equation 4.86 are deduced by using Equation 4.84 instead of Equation 4.83. For $\sigma_2 = (1 + \delta)\sigma_1$ a term of the order $1/3 \delta^3$ is neglected.

4. Physics

The convolution of two TPNDs is, in general, *not* a TPND. The mean value F , variance V and third moment M are additive for a convolution. The convolution of a TPND f_1 with moments F_1 , V_1 and M_1 and a TPND f_2 with moments F_2 , V_2 and M_2 results in a function f with moments

$$\begin{aligned} F &= F_1 + F_2 \\ V &= V_1 + V_2 \\ M &= M_1 + M_2. \end{aligned} \tag{4.90}$$

For not too large asymmetries (with the difference between σ_1 and σ_2 being less than about 50%), the convolution of two TPNDs is very close to a TPND, and f can be approximated by a TPND with moments F , V and M . The parameters of this TPND are then given by [Equation 4.85](#), [Equation 4.86](#) and [Equation 4.87](#).

4.10.2. Asymmetry parameter of arbitrary distributions

A possible measure of the skew of a distribution is its third standardized moment, the skewness $\gamma = M/V^{3/2}$. However, the skewness can be derived only with large errors from experimental data or from data obtained by Monte-Carlo computer simulations due to the large influence of small fluctuations in the wings of the distributions. Some textbooks generally discourage to use the skewness for experimental data [92]. Moreover, angular and energy distributions due to multiple small-angle scattering have infinite variance and an undefined third moment [109], so that the skew of these distributions cannot be measured by γ . The problem of infinite variance was overcome in [109] by using the full width at half maximum (FWHM) instead of using the variance V . Following this way, we introduce the asymmetry parameter α as a measure of the skew of a distribution, with

$$\alpha = \frac{h_2 - h_1}{h_1 + h_2}. \tag{4.91}$$

h_2 is the half width at half maximum (HWHM) towards higher values, i.e. the distance between the value at which the distribution has decreased to its half maximum and the mode. h_1 is the HWHM towards low values. α can be determined for any unimodal distribution, even if V or M are infinite or undefined, and can be determined with sufficient accuracy for experimental or Monte-Carlo data. Consequently we use the FWHM instead of the variance and the asymmetry parameter α instead of the skewness γ for all comparisons to experimental or Monte-Carlo data.

4.10.3. Electronic energy-loss straggling

For small energy losses the energy distributions of slowing-down ions in matter are non-Gaussian with long tails towards low energies [118]. For larger energy losses the increasing number of statistically independent collisions tends to draw the energy distributions near to a Gaussian distribution. The energy distributions are typically almost Gaussian for energy losses in the range of 10–30% of the primary energy, and their widths can be described by the Bohr theory [25], but have to be corrected for electron binding [35] and charge-state fluctuations [128] in order to achieve a sufficient accuracy. For energy losses exceeding about 30% of the primary

energy the widths and shapes of the energy distributions are determined by non-stochastic broadening or squeezing due to the shape of the stopping power [91, 115–117].

As was already shown in [116], the first three moments are sufficient for an accurate approximation of the energy-loss distributions if the energy-loss is not too small. We obtain the mean value from the energy loss due to the stopping power [6, 61, 132, 135, 136] and the variance from a straggling theory, such as [128]. As third moment we use nonrelativistic free Coulomb scattering [99, chapter 8.11]. Propagation of the HWHM's in thick layers is taken approximately into account by calculating the energy losses of the mode energy E_0 and the energies $E_0 - h_1$ and $E_0 + h_2$ during a step Δx . No information is available about the third moment of charge state fluctuations [128]: These are assumed to be Gaussian.

The Γ -distribution was proposed in [18] for modeling the shape of the energy-loss distribution. However, because the Γ -distribution has only two adjustable parameters (mean value and variance), the third moment cannot be adjusted and will usually not coincide with the correct third moment of the distribution: The method proposed in [18] therefore generally will not approximate correctly the energy-loss distribution.

The straggling of 19.68 MeV protons in aluminum is shown in Figure 4.19. The experimental data were digitized from [117, Fig. 3], their half widths and asymmetry parameters were obtained by fitting. At this high incident energy multiple scattering is negligible, so that the shape of the energy distributions is determined by electronic energy-loss straggling only. The incident beam energy spread was 40 keV FWHM [117] and is assumed to be Gaussian. The calculated half-widths of the energy distributions are in excellent agreement with the measurements. Bohr straggling describes the half-widths accurately at energies above about 15.5 MeV, i.e. for energy losses below about 20% of the primary energy. At larger energy losses the half-widths are larger than Bohr straggling due to straggling propagation. The asymmetry parameter, as shown in Figure 4.19 bottom, is also in very good agreement with the experimental values. At very large energy losses (in depths above about 0.5 g/cm²) the model somewhat underestimates the asymmetry.

The measured energy distributions of 19.68 MeV protons after traversing 0.2675 g/cm² and 0.497 g/cm² aluminum are shown in Figure 4.20. The energy distributions are well approximated by the TPNDS. The approximation by Gaussian energy distributions gives the correct widths (FWHMs) of the distributions, but the shape is not as well reproduced.

4.10.4. Multiple scattering

Szilágyi et al. [109] and Amsel et al. [2] have developed an analytical theory of multiple small angle scattering, which has been implemented in the computer codes DEPTH [109] and SIMNRA [72, 77]. This analytical theory intrinsically assumes symmetry of all angular and energy spread distributions, which are approximated by (symmetric) Pearson VII distributions. The theory also neglects energy transfer to target atoms, i.e. nuclear straggling [77]. As has been already shown by comparison to Monte-Carlo simulation calculations [77], this theory correctly predicts the shape and FWHM of angular spread distributions: These are (almost) symmetric until large energy losses and can be described with sufficient accuracy by Pearson VII distributions. This theory also predicts the FWHM of energy spread distributions due to multiple scattering with sufficient accuracy [77]. It, however, does not describe correctly the shape of energy spread

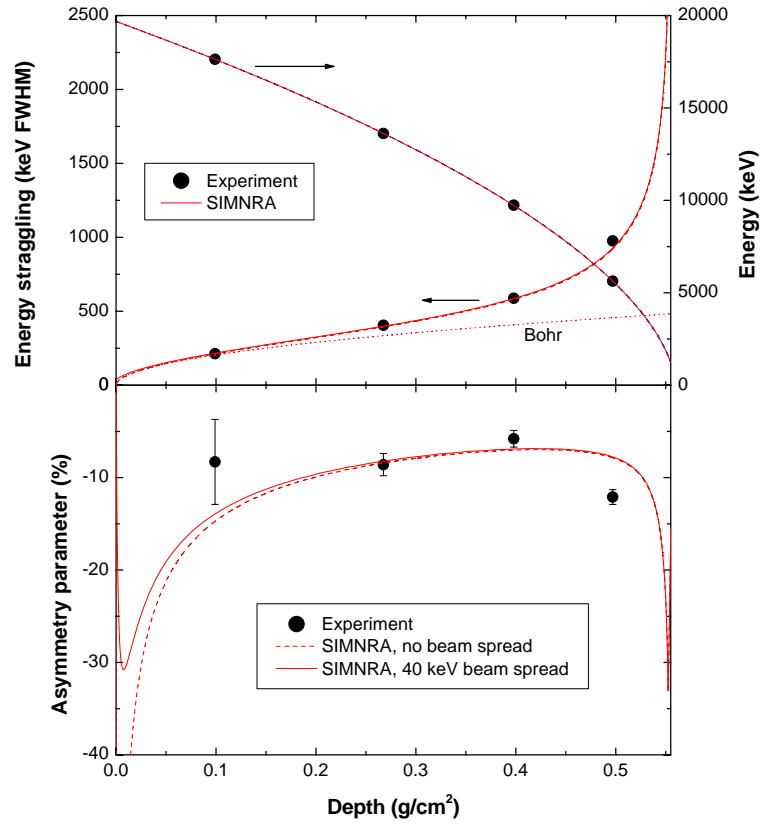


Figure 4.19.: Straggling of 19.68 MeV protons in aluminum. Top: Mode energy and width (FWHM) of the straggling distributions. Bottom: Asymmetry parameter (see Equation 4.91) of the straggling distributions. Experimental data from [117]. Solid lines: 40 keV Gaussian energy spread of incident beam; Dashed lines: Without energy spread of incident beam.

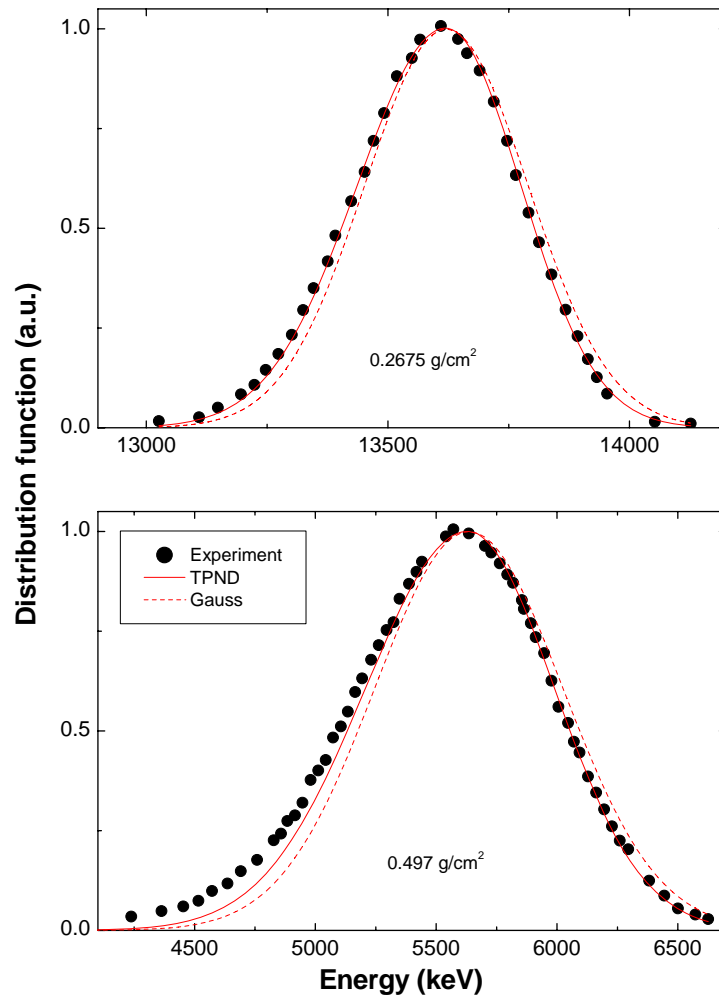


Figure 4.20.: Energy distributions of 19.68 MeV protons after traversing 0.2675 g/cm² and 0.497 g/cm² aluminum. Dots: Experimental data from [117]; Solid line: SIMNRA calculation using a two-piece normal distribution; Dashed line: SIMNRA calculation using a Gaussian energy distribution.

4. Physics

distributions: These are not symmetric, but always skewed towards low energies [77]. This skewness has two main reasons: Path length differences of particles scattered to different sides and nuclear straggling.

The influence of path length differences is shown schematically in Figure 4.21: At oblique incidence particles scattered to one side (trajectory A) on average have a shorter total path length (and corresponding energy loss) until they reach a specific depth d than particles scattered to the other side (trajectory B). The path length l_A for trajectory A is given by $l_A = d/\cos(\alpha - \Delta\epsilon/2)$, with α the angle of incidence and $\Delta\epsilon$ the chord angle (see Figure 4.21). The path length l_B for trajectory B is given by $l_B = d/\cos(\alpha + \Delta\epsilon/2)$. For identical chord angles to both sides this results in an asymmetric distribution of path lengths and a corresponding asymmetric energy distribution. Path length differences play especially a role at grazing incidence and for heavy target elements.

Multiple scattering of 2 MeV ^4He in Au at 60° angle of incidence, as calculated with the Monte-Carlo code MCERD and by SIMNRA, is shown in Figure 4.22. The widths of the angular distributions (Figure 4.22 top) are in excellent agreement between both codes. The widths of the energy spread distributions (Figure 4.22 middle) are in excellent agreement until a depth of about 500 nm. At larger depths SIMNRA overestimates the width. The asymmetry parameter α (Figure 4.22 middle, see Equation 4.91 for the definition of α) is in agreement between MCERD and SIMNRA within the error bars. The large error bars for the MCERD data are mainly due to the fit inaccuracy for the position of the mode of the energy distributions, which has a large influence on the determination of α . The calculated energy distributions in depths of 100 nm, 300 nm and 500 nm are shown in Figure 4.23. At small energy losses (100 nm) the distribution is almost symmetric with tails towards lower and higher energies. These tails are characteristic for multiple scattering [2] and cannot be described by Gaussian-type functions. At larger energy losses (300 nm and 500 nm) the distributions get more and more asymmetric. The high-energy parts of the distributions approach a Gaussian distribution, while the low-energy parts develop broad tails extending to very low energies. The body of these distributions can be well described by a TPND, which gives a better approximation to the Monte-Carlo results than a Gaussian. The low-energy tail is mainly due to energy transfer to target atoms, i.e. nuclear straggling. Energy transfer to target atoms is neglected in the Szilágyi/Amsel theory of multiple scattering [2, 109], and nuclear straggling at moderate energy losses is so strongly skewed that it cannot be described by a skewed Gaussian-type function. A successful description of this low-energy tail therefore would require an additional convolution, which is very costly in terms of computing time. This is especially important for heavy ions, where the nuclear energy-loss straggling contribution may be large and cannot be approximated by any skewed Gaussian type of function.

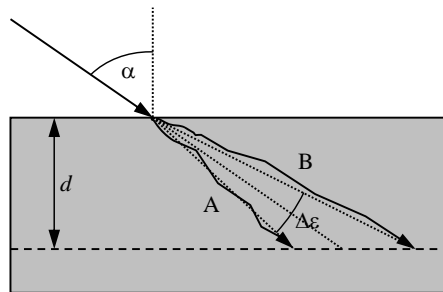


Figure 4.21.: Schematic representation of multiple scattering at oblique incidence. Depth d and angle of incidence α . Trajectories A and B are assumed to have chord angles of $\mp\epsilon/2$, with the FWHM of the chord angle distribution $\Delta\epsilon$.

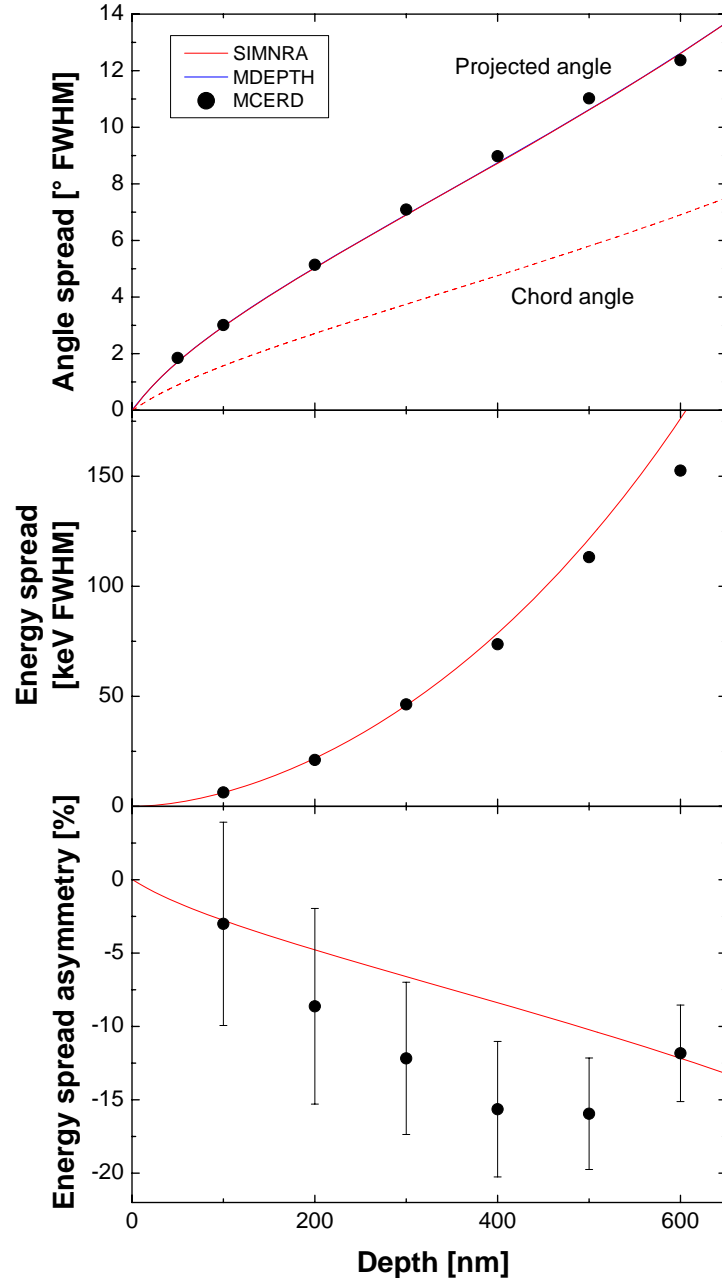


Figure 4.22.: Multiple scattering of 2 MeV ^4He in gold at 60° angle of incidence. Top: FWHMs of the angular spread distribution $\Delta\Phi$ and the chord angle distribution $\Delta\epsilon$; middle: FWHM of the energy spread distribution; bottom: Asymmetry parameter α of the energy spread distribution. Solid dots: Calculation with the Monte-Carlo code MCERD; lines: Calculation with SIMNRA. The MDEPTH calculation for the angular spread distribution coincides with the SIMNRA result.

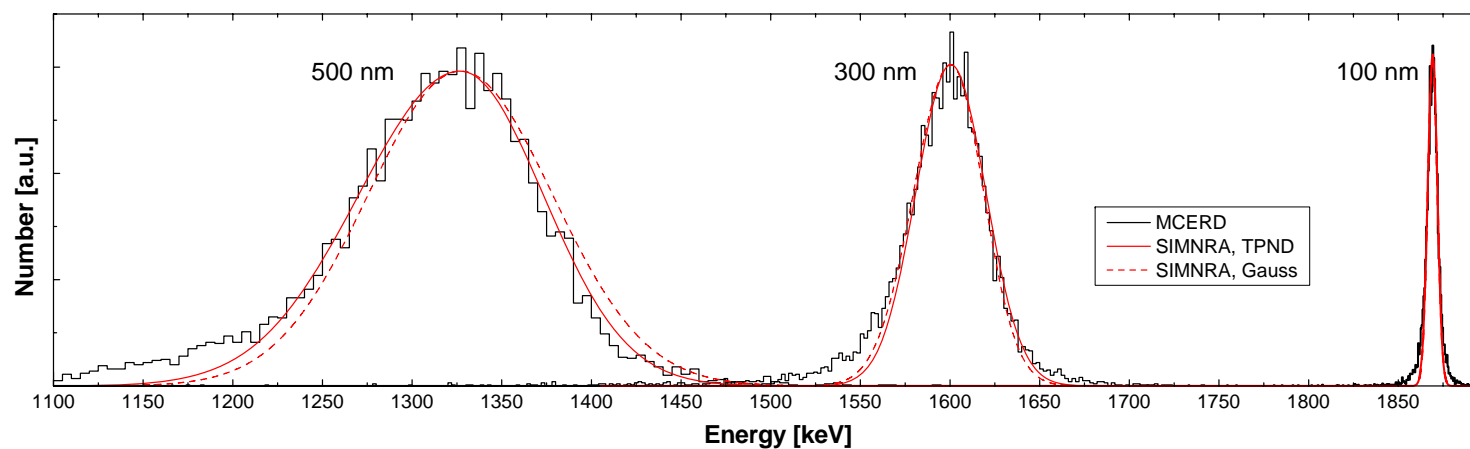


Figure 4.23.: Energy distributions due to multiple scattering of 2 MeV ^4He in gold at 60° angle of incidence. in depths of 100 nm, 300 nm and 500 nm. Histograms: Calculation with the Monte-Carlo code MCERD; dashed lines: Calculation with SIMNRA assuming a Gaussian shape of the distributions; solid lines: Calculation with SIMNRA assuming a two-piece normal distribution (TPND) shape of the distributions.

4.11. Surface roughness

The quantitative application of ion beam analysis methods is usually restricted to laterally homogeneous and smooth films. But the experimentalist is often confronted with rough surfaces. The effects of rough surfaces of *thick* targets on RBS were investigated in some detail by Edge and Bill [43], Knudson [60], Bird *et al.* [23], and Hobbs *et al.* [55]. Wüest and Bochsler [127] and Yesil *et al.* [129, 130] attacked the problem by means of a Monte-Carlo computer simulation, taking into account correlation effects of the surface roughness and multiple surface crossings of the incident and emerging ions. It turned out that effects of rough surfaces of *thick* targets occur only for grazing angles of the incident or emerging ions. This is especially the case in ERDA on thick, rough targets, as was shown by Yesil *et al.* [129, 130] and Kitamura *et al.* [59]. Hydrogen depth profiling on rough surfaces by ERDA was studied experimentally by Behrisch *et al.* [20].

Astonishingly, the effects of rough *thin* films were studied much more scarcely. For RBS, rough films on a smooth substrate were investigated by Shorin and Sosnin [98] and Metzner *et al.* [83, 84]. Shorin and Sosnin [98] used a Monte-Carlo computer simulation. The Monte-Carlo approach suffers from long computing times of the order of hours [130], rendering these codes impractical for evaluation of experimental spectra. Moreover, the Shorin/Sosnin code treats only RBS with Rutherford cross-sections, neglecting non-Rutherford scattering, NRA and ERDA. The theoretical approach of Metzner *et al.* [83, 84] allows to extract the thickness distribution of rough films from a measured spectrum. However, this approach is only valid for RBS with Rutherford cross sections, a scattering angle of exactly 180° and constant stopping power, thus severely limiting the practical applicability of this work. The computer code RUMP [40, 41] allows to fuzz the interface between two layers by roughening the top layer. However, this is intended only for small roughness amplitudes, and the roughness distribution function is not documented.

Moreover, all work done so far treats only the case of a rough film on a smooth substrate. But in practice also the case of a film deposited on a rough substrate (Figure 4.24(right)) is sometimes encountered. This section describes the algorithms used for the description of rough surfaces and compares results of code calculations with experimental data. The limitations of the used approximations are discussed.

4.11.1. Rough film on a smooth substrate

A rough film on a smooth substrate is shown schematically in Figure 4.24(left). The substrate can be considered to be smooth, if its roughness is much smaller than the mean thickness \bar{d} of the film. The film thickness distribution is described by a distribution function $p(d)$, with the film thickness d measured perpendicular to the substrate, see Figure 4.24(left) and $d \geq 0$. In the literature, usually a Gaussian distribution centered at \bar{d} with variance σ^2 and cut-off at zero is used for $p(d)$ [83, 84]. However, a more natural choice of a distribution function with only positive values $d \geq 0$ is the Gamma distribution, which is also fully described by its mean value \bar{d} and standard deviation σ . The Gamma distribution is defined by

$$p(d) = \frac{\beta^\alpha}{\Gamma(\alpha)} d^{\alpha-1} e^{-\beta d}, d > 0 \quad (4.92)$$

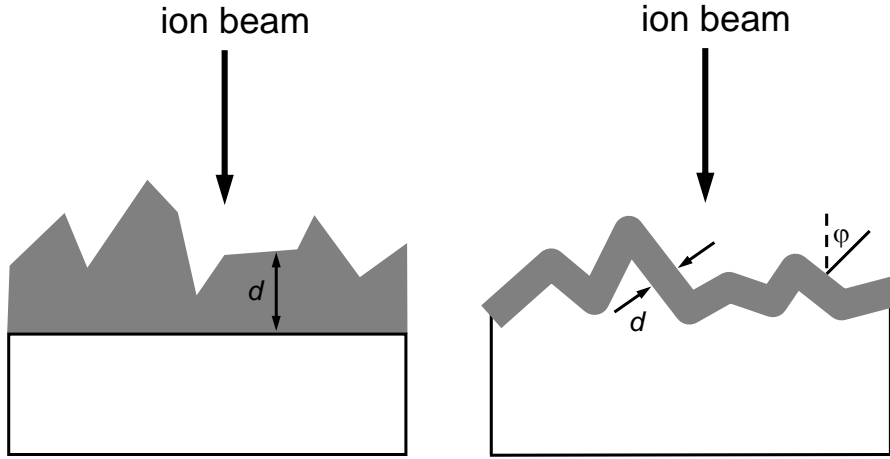


Figure 4.24.: Schematic representation of a rough film on a smooth substrate (left), and of a smooth film on a rough substrate (right).

with $\alpha = \bar{d}^2/\sigma^2$ and $\beta = \bar{d}/\sigma^2$. $\Gamma(\alpha)$ is the Gamma function. The Gamma distribution is shown in Figure 4.25 for $\bar{d} = 1$ and different standard deviations σ . The corresponding Gaussian distributions centered at 1 and identical σ are shown for comparison. If $\sigma \ll \bar{d}$, i.e. if the width of the distribution is small compared to its mean value, Gaussian and Gamma distributions are nearly identical, see the curves for $\sigma = 0.1$ in Figure 4.25. With increasing σ the two distributions get more and more different (see the curves for $\sigma = 0.3$ and 0.7 in Figure 4.25). For $\sigma = \bar{d}$ the Gamma distribution decreases exponentially with $p(d) = e^{-d}$, and for $\sigma > \bar{d}$ an integrable singularity develops at $d = 0$.

A RBS, NRA or ERDA spectrum of a rough film is approximated by a superposition of N spectra with different layer thicknesses d_i . N can be adjusted by the *Number of thickness steps* in the *Setup: Calculation...* menu, see subsection 3.5.3. Typically about $N = 20$ sub-spectra are necessary to obtain a smooth superposition, though N has to be increased to about $N = 50$ for broad distributions with $\sigma \geq \bar{d}$. The weight w_i of each sub-spectrum is determined according to the thickness distribution function. For each sub-spectrum the layer is treated to be smooth with thickness d_i . Correlation effects, such as incidence through a hump and emergence through a valley or multiple surface crossings, are neglected. This is only correct for backscattering at a scattering angle of exactly 180° and for transmission geometries. However, for scattering angles in the range $150^\circ - 180^\circ$ and non-grazing incidence and emergence angles, as are used in many RBS and NRA setups, correlation effects still play only a minor role and can be neglected without severe loss of accuracy. But it should be kept in mind that the used approximation gets invalid for grazing incidence or exit angles, as is the case in ERDA - in these cases correlation effects may be dominant and can change the shape of the spectra considerably.

The effect of layer roughness on the shape of RBS spectra is shown in Figure 4.26 for incident ^4He ions backscattered from a gold layer at a scattering angle of 165° . The film thickness distributions are described by the Gamma distributions shown in Figure 4.25. If the thickness variation is much smaller than the mean film thickness ($\sigma/\bar{d} = 0.1$), only the low energy edge of

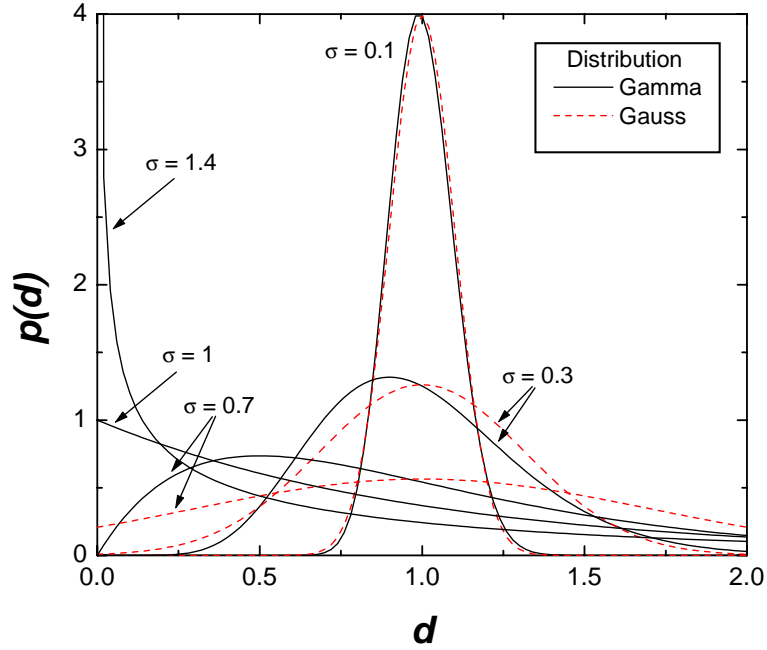


Figure 4.25.: Comparison of Gaussian distribution functions centered at 1 (dashed lines) and Gamma distribution functions (solid lines) with mean value $\bar{d} = 1$ and different standard deviations σ .

the film is affected by the roughness and gets broader. With increasing roughness the broadening of the low energy edge increases, until at $\sigma/\bar{d} \approx 0.6$ the high energy edge begins to decrease. The energy $E_{1/2}$, at which the low energy edge has decreased to its half height, remains fairly constant until large roughness amplitudes of the order $\sigma/\bar{d} \approx 0.6$, i.e. until the high energy edge begins to decrease. For sufficiently thick films, i.e. if the film is completely resolved, this energy is therefore a rather robust measure of the mean film thickness even for large roughnesses, as long as the high energy edge is not affected.

The energy spectrum of 1.5 MeV ^4He backscattered from a rough Ni-film deposited on polycrystalline carbon is shown in Figure 4.27. The measured spectrum is well reproduced in the simulation by a mean Ni layer thickness of 2.17×10^{18} Ni-atoms/cm² (238 nm) and a roughness with standard deviation $\sigma = 2.12 \times 10^{17}$ Ni-atoms/cm² (23 nm) (solid line). The experimental data are not well reproduced by the spectrum of a smooth Ni layer (dashed line). The roughness of the Ni film was determined from line scans with a profiler. The roughness distribution, i.e. the deviation of the actual surface from the leveled one, was approximately Gaussian: For small values of σ/\bar{d} a Gaussian and a Gamma distribution cannot be distinguished, see Figure 4.25. The carbon substrate was already rough with a standard deviation $\sigma_C = 18.2$ nm. The roughness of the Ni film on the substrate was $\sigma_{C+Ni} = 26.5$ nm. This roughness is made up by the roughness of the carbon substrate plus the roughness of the Ni film σ_{Ni} . By assuming the two roughnesses to be independent, i.e. $\sigma_{C+Ni}^2 = \sigma_C^2 + \sigma_{Ni}^2$, the roughness of the Ni film alone is about 19.3 nm, in good agreement with the result from He backscattering of 23 nm.

The energy spectrum of 2.0 MeV ^4He backscattered from a rough oxidised aluminum film on polycrystalline carbon is shown in Figure 4.28. The carbon substrate was well polished and had

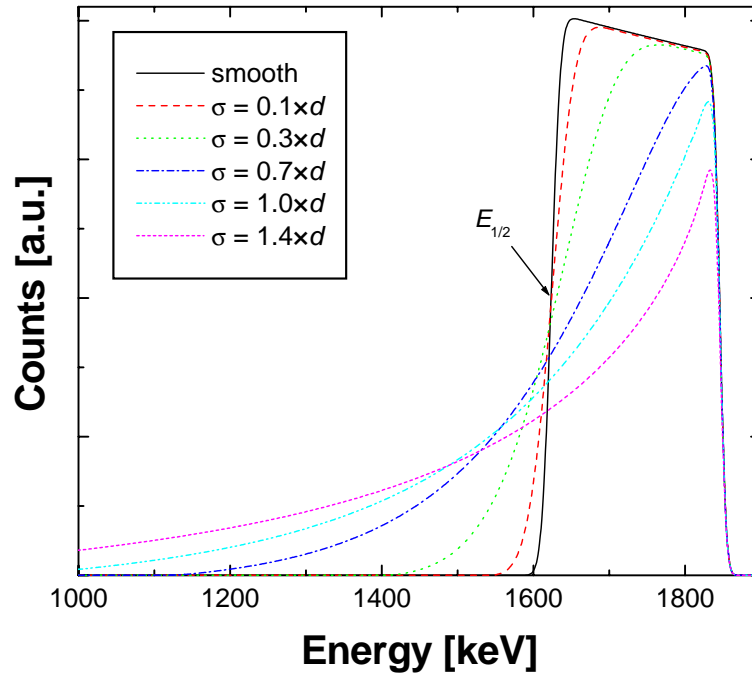


Figure 4.26.: Calculated energy spectra for 2 MeV ^4He backscattered from a smooth and rough gold layers with mean thickness $\bar{d} = 1 \times 10^{18}$ Au-atoms/cm 2 and different roughnesses with standard deviation σ . The film thickness distributions are shown in Figure 4.25. Incident angle $\alpha = 0^\circ$, scattering angle 165° . $E_{1/2}$ marks the energy, at which the low energy edge has decreased to its half height.

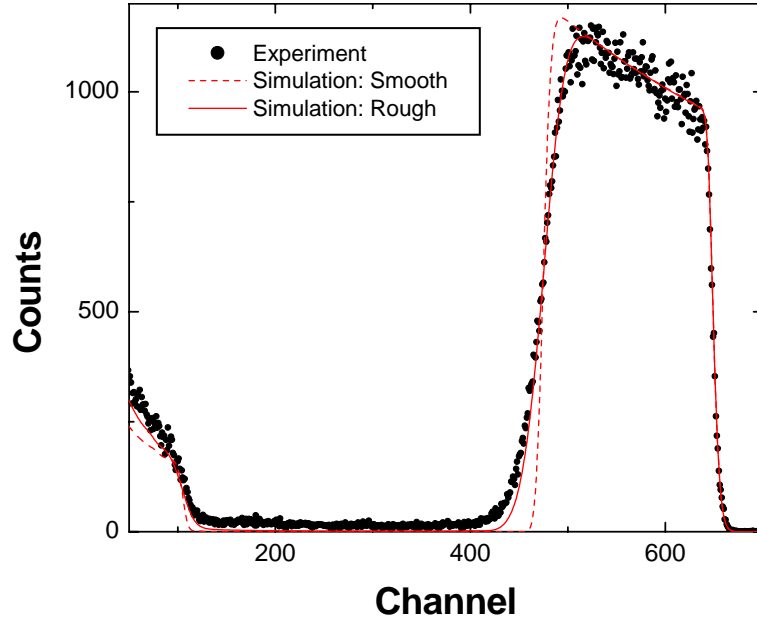


Figure 4.27.: 1.5 MeV ^4He backscattered at 165° from a rough Ni-film with a mean thickness of 2.17×10^{18} Ni-atoms/cm 2 on carbon substrate. Dots: Experimental data; Dashed line: Simulation assuming a smooth Ni layer; Solid line: Simulation assuming a rough Ni layer with roughness $\sigma = 2.12 \times 10^{17}$ Ni-atoms/cm 2 .

a mean roughness < 25 nm [80]. The film was exposed for about 8 months as erosion monitor at the vessel wall of the nuclear fusion tokamak experiment JET [79, 80], the wall temperature was about 300°C . The initial Al layer thickness was 3.16×10^{18} atoms/cm 2 (525 nm), but decreased due to sputtering by bombardment with energetic hydrogen atoms from the nuclear fusion plasma to 7.5×10^{17} Al-atoms/cm 2 . At the same time the Al film was oxidised and some nickel, which was initially eroded at an erosion dominated area of the JET vessel wall⁸, was redeposited on the Al film and incorporated. The observed spectrum with the tails at the low energy sides of the O, Al and Ni peaks cannot be reproduced by assuming a smooth layer. But it is fairly well reproduced by a rough layer with a mean film thickness of 1.11×10^{18} atoms/cm 2 , roughness $\sigma = 1.06 \times 10^{18}$ atoms/cm 2 , and composition 68% Al, 30% O, 2% Ni (solid line in Figure 4.28). The shape of the film thickness distribution is close to the curve with $\sigma = 1$ in Figure 4.25. This example shows clearly, that non-Gaussian distributions of layer thicknesses are observed in practice and can be described by a Gamma distribution.

4.11.2. Smooth film on a rough substrate

A film with homogeneous thickness d on a rough substrate is shown schematically in Figure 4.24(right). The thickness d is measured parallel to the local surface normal. If the film consists of several layers, then the layers are stacked on each other and follow the rough surface of the substrate, see Figure 4.29. The thicknesses d_1, d_2, \dots of layers 1, 2, ... are measured parallel to the local surface normal.

⁸The JET vessel walls consist of Inconel, a stainless steel with high nickel content.

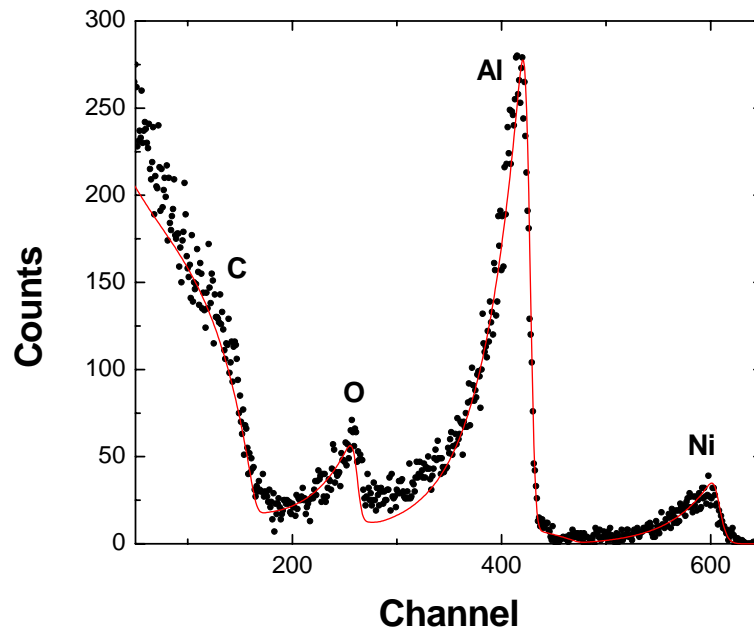


Figure 4.28.: 2 MeV ^4He backscattered at 165° from a rough oxidised aluminum film on carbon. The film was used as long term sample in the tokamak JET and was strongly eroded by plasma impact. Additionally some Ni was deposited from the plasma. Dots: Experimental data; Solid line: Simulation with a mean film thickness of 1.11×10^{18} atoms/cm 2 and roughness $\sigma = 1.06 \times 10^{18}$ atoms/cm 2 . Film composition 68% Al, 30% O, 2% Ni.

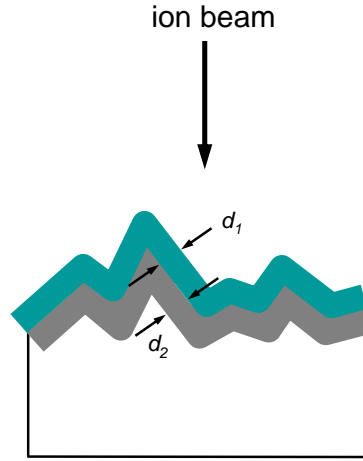


Figure 4.29.: Schematic representation of a smooth multi-layered film on a rough substrate. d_1 is the thickness of layer 1, d_2 the thickness of layer 2.

The substrate is considered to be rough, if its roughness amplitude is much larger than the thickness d of the film. We assume a rough substrate to consist of inclined line segments with local inclination angle φ (see Figure 4.24(right) and Figure 4.31), and the film thickness d is measured parallel to the local surface normal. Such a rough surface is described by a distribution of local tilt angles $p(\varphi)$. The concept of a local tilt angle was already used by Küstner *et al.* for the calculation of the sputtering yield of rough surfaces by ion bombardment in the energy range 100 eV to several keV [65]. In Küstner's work the rough surface was treated as a fully 3-dimensional object, which was necessary due to the 3-dimensional nature of the collision cascades created by keV ions. In MeV ion beam analysis the trajectories of the incident and emerging ions can be approximated with good accuracy by straight lines, and we have to consider only the intersection of the plane, which is formed by the directions of the incident and emerging ions, and the target surface, see Figure 4.30: This is only a 2-dimensional line profile as the one shown in Figure 4.24(right).

The tilt angle distribution is given by $p(\varphi)$. This distribution describes the frequency of occurrence of a line segment inclined by φ . A rough surface without preferential orientation has a mean tilt angle

$$\bar{\varphi} = \int_{-90^\circ}^{90^\circ} \varphi p(\varphi) d\varphi = 0^\circ. \quad (4.93)$$

The probability distribution $\tilde{p}(\varphi)$ of hitting a surface tilted by φ by an incident ion is given by

$$\tilde{p}(\varphi) = p(\varphi) \cos(\alpha - \varphi), \quad (4.94)$$

with the incident angle α of the ion. α is measured towards the surface normal of a non-inclined surface. The factor $\cos(\alpha - \varphi)$ is due to the projection of the line segment into the plane perpendicular to the incident ion trajectory: It is more likely to hit a segment which is perpendicular to the incident trajectory than an inclined segment - and obviously it is impossible to hit a segment which is tilted parallel to the incident beam. It is important to note that a

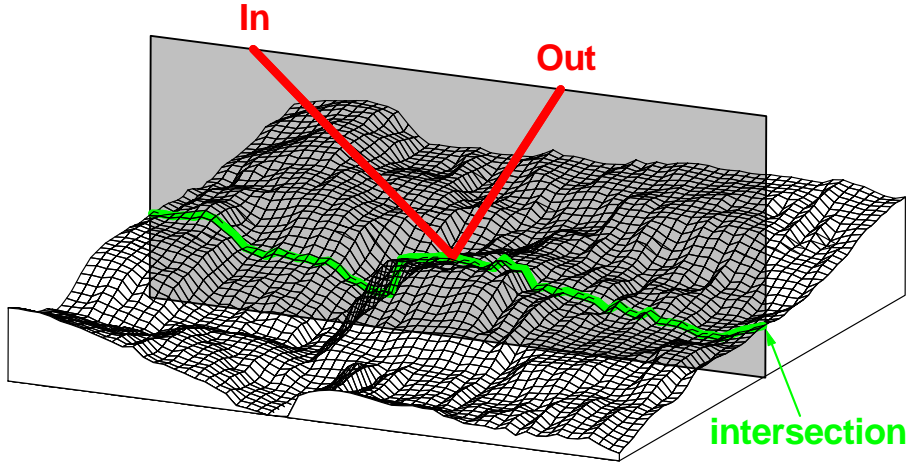


Figure 4.30.: Schematic representation of a rough surface. In: Direction of the incident beam; Out: Direction of the outgoing beam; Light gray: Plane spanned by the incident and outgoing beams; Intersection: Intersection of this plane with the rough surface.

profiler or a scanning tunneling microscope (STM), which samples the surface at a constant step width parallel to the surface, measures the distribution $\tilde{p}(\varphi)$ rather than $p(\varphi)$: Large tilt angles are under-represented, and tilt angles of 90° cannot be measured at all.

RBS, NRA and ERDA spectra of a smooth film on a rough substrate are approximated by a superposition of M spectra with different local incident angles $\tilde{\alpha}$, where

$$\tilde{\alpha} = |\alpha - \varphi| \quad (4.95)$$

and local exit angles $\tilde{\beta}$. The choice of $\tilde{\beta}$ is discussed below. M can be adjusted by the *Number of angular steps* in the *Setup: Calculation...* menu, see subsection 3.5.3. The weight of each sub-spectrum is determined according to the distribution function $\tilde{p}(\varphi)$. For each sub-spectrum the substrate is treated to be smooth, i.e. a spectrum for a smooth layer, but with angles $\tilde{\alpha}$ and $\tilde{\beta}$ is calculated. Incident angles $\tilde{\alpha} > 90^\circ$ are excluded: This represents a line segment which cannot be hit by the incident beam. As in the case of a rough film on a smooth substrate, surface correlation effects like shadowing of one line segment by another, and multiple surface crossings are neglected.

In IBM geometry (with $\alpha + \beta + \theta = 180^\circ$), the local exit angle $\tilde{\beta}$ for a given inclination angle φ is given by

$$\tilde{\beta} = |\beta + \varphi| \quad (4.96)$$

This relation was used by SIMNRA 4.70–4.90 and ref. [74]. But this choice of β is only valid in IBM geometry and results in unrealistic spectra if a rough surface is bombarded at non-normal incidence in non-IBM geometry. SIMNRA 5.00–5.20 used a more general approach, where the correct relation between the angles in any geometry and even the 3-dimensional nature of a rough surface were taken into account by rotating the inclined line segments and calculating an averaged β . Although more accurate than Equation 4.96, it turned out that this approach suffered from numerical instabilities at non-normal incident angles $\alpha > 30^\circ$ and non-IBM

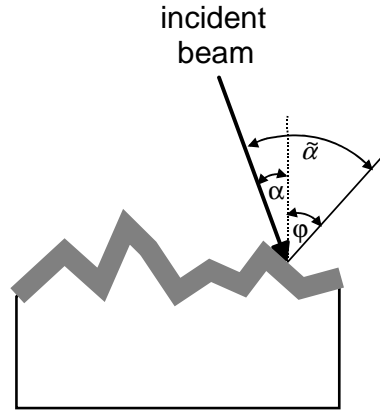


Figure 4.31.: Schematic representation of the incident angle α , local inclination angle φ , and local incident angle $\tilde{\alpha}$.

geometry. In SIMNRA 5.25 the choice of $\tilde{\beta}$ was changed again and is calculated in the following way:

1. For each inclination angle the local angle of incidence $\tilde{\alpha}$ is calculated according to [Equation 4.95](#).
2. The azimuth angle ψ between the incident and exit beams is calculated from

$$\sin \psi = \frac{\cos \beta + \cos \theta \cos \alpha}{\sin \theta \sin \alpha}. \quad (4.97)$$

3. As scattering angle θ and azimuth angle ψ are constant for a given geometry, the local exit angle $\tilde{\beta}$ can be calculated for a given local incident angle $\tilde{\alpha}$ from

$$\cos \tilde{\beta} = \sin \psi \sin \theta \sin \tilde{\alpha} - \cos \theta \cos \tilde{\alpha}. \quad (4.98)$$

[Equation 4.98](#) is valid for any geometry.

For a given local tilt angle φ , the calculation of the local incident and exit angles $\tilde{\alpha}$ and $\tilde{\beta}$ depends on the setting of the *Dimension of substrate roughness* switch, see [subsubsection 3.5.3](#).

- *2-dimensional:* In the 2-dimensional model the local incident and exit angles $\tilde{\alpha}$ and $\tilde{\beta}$ are calculated according to [Equation 4.95](#) and [Equation 4.98](#).
- *2.5-dimensional:* In the 2.5-dimensional model it is assumed, that the surface consists of small planes, which are tilted by a local tilt angle φ , and rotated by an angle ω along the surface normal. In a fully 3-dimensional surface model, the local incident and exit angles $\tilde{\alpha}$ and $\tilde{\beta}$ depend on φ and ω , and it would be necessary to calculate spectra for each possible combination of φ and ω . In order to keep computing times short, this is not done. Instead, for each tilt angle the average angles $\tilde{\alpha}$ and $\tilde{\beta}$ are used, averaged over ω . Therefore the 2.5-dimensional model uses only one set $\tilde{\alpha}, \tilde{\beta}$ for each φ (as would be

the case in the 2-dimensional case), but this set is obtained by averaging a 3-dimensional surface model. The accuracy of the 2.5-dimensional model is therefore somewhere in between a 2-dimensional and a fully 3-dimensional model.

Which distribution has to be used as tilt angle distribution? A line profile of the surface of a carbon fibre composite (CFC) material manufactured by Dunlop is shown in [Figure 4.32](#) (top). Due to its high thermal conductivity this material is used for high heat flux components in the tokamak experiment JET. The standard deviation of the surface roughness is about $\sigma = 8.2 \mu\text{m}$. A histogram of the tilt angle distribution $\tilde{p}(\varphi)$, obtained with a profiler from several line scans in different sample directions, is shown in [Figure 4.32](#) (bottom). Tilt angles larger than about 60° are not observed due to the apex angle of the profiler tip. The tilt angle distribution is fairly well described by a Lorentz distribution for the tilt angles $p(\varphi)$ (dashed line), while a Gaussian distribution underestimates strongly the wings of the distribution (dotted line). The full width at half maximum (FWHM) of the Lorentz distribution is 26.6° .

Calculated backscattering spectra for incident ^4He ions backscattered from a gold layer with thickness $1 \times 10^{18} \text{ atoms/cm}^2$ at a scattering angle of 165° are shown in [Figure 4.33](#) for a smooth and rough substrates. The rough substrates are described by a Lorentz distribution of tilt angles with different FWHM w . On a rough substrate the low energy edge gets a tail, which increases with increasing roughness. This tail extends to energies close to zero. With increasing roughness the Au peak also gets broader, and the energy $E_{1/2}$, at which the low energy edge has decreased to its half height, is no good measure for the film thickness: It depends on the roughness of the substrate. The high energy edge and the plateau are only slightly affected by substrate roughness and decrease only little at large roughnesses due to shadowing: The backscattered particles do not reach the detector any more, because the exit angle β points inside the layer. For $w = \infty$ the local tilt angles are equipartitioned, and the corresponding spectrum represents the case of maximum roughness.

A measured spectrum for 2.5 MeV protons backscattered from a tungsten layer on top of a rough carbon substrate is shown in [Figure 4.34](#). The non-Rutherford elastic scattering data from [1] were used for the C(p,p)C cross section. The substrate is the same CFC material, which surface is shown in [Figure 4.32](#). The mean W layer thickness was about $3.5 \mu\text{m}$, while the standard deviation of the substrate roughness was about $8.2 \mu\text{m}$, i.e. the substrate roughness was considerably larger than the thickness of the W layer. The dotted line in [Figure 4.34](#) is the calculated spectrum for a smooth W layer on a smooth carbon substrate. Plural scattering in the W layer was included in dual scattering approximation, see [subsection 4.9.3](#). Plural scattering results for example in the small background visible between the carbon and tungsten signals in channels 500–650. This spectrum has only minor resemblance with the experimental curve, and requires a slightly thicker W layer ($3.6 \mu\text{m}$) for best fit. The dashed line is calculated for a rough W layer, characterized by a Gamma-distribution of layer thicknesses with a mean thickness of $3.5 \mu\text{m}$ and standard deviation $\sigma = 0.3 \mu\text{m}$ on a rough carbon substrate, characterized by a Lorentz distribution of tilt angles with FWHM $= 20^\circ$. The roughnesses of the layer and the substrate are assumed to be independent, and plural scattering is not taken into account. The W peak (channels > 650) is already well described, but the low energy tail below the peak is underestimated. The solid line uses the same roughness parameters, but takes additionally plural scattering into account. Now the whole experimental spectrum is reproduced well. Compared

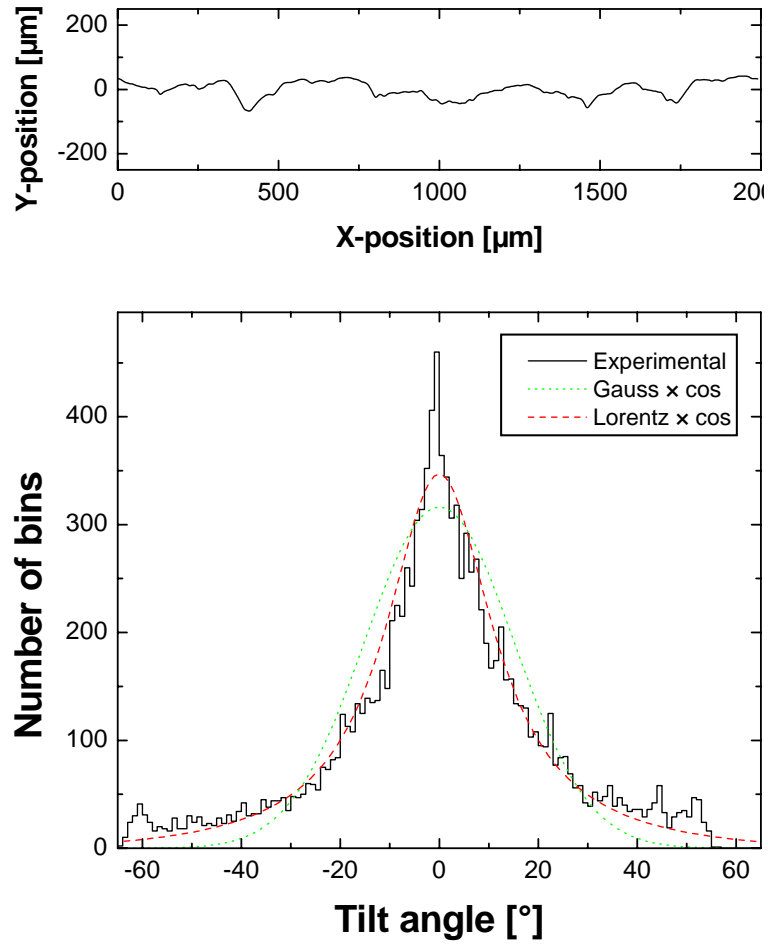


Figure 4.32.: Top: Line profile of a carbon fibre composite (CFC) surface. Bottom: Histogram of the local tilt angle distribution of the CFC surface. Solid line: Experimental data; Dashed line: Lorentz distribution times cosine of the tilt angle; Dotted line: Gaussian distribution times cosine of the tilt angle.

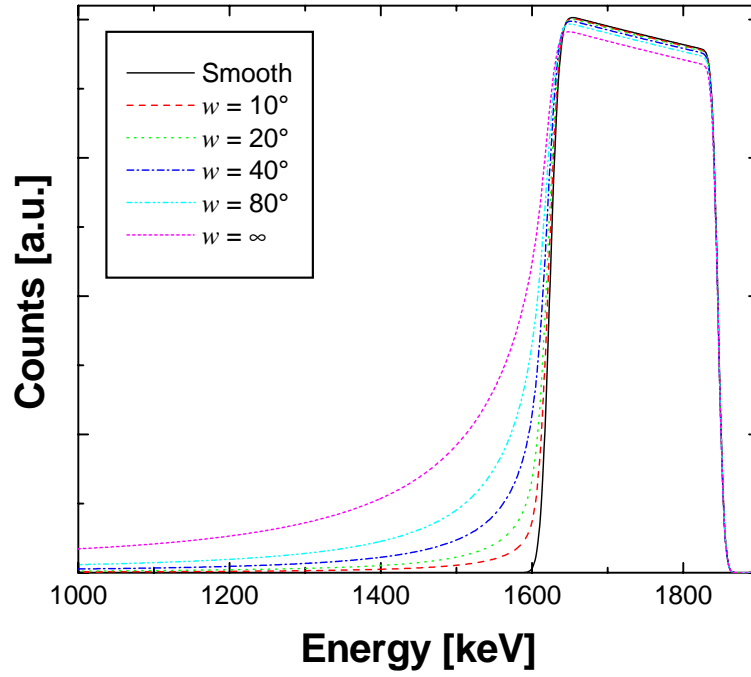


Figure 4.33.: Calculated energy spectra for 2 MeV ^4He backscattered from a gold layer with thickness 1×10^{18} Au-atoms/cm 2 on a rough substrate with different roughnesses. The roughness is described by a Lorentz distribution of tilt angles with FWHM w . $w = \infty$ is an equipartition of tilt angles. Incident angle $\alpha = 0^\circ$, scattering angle 165° .

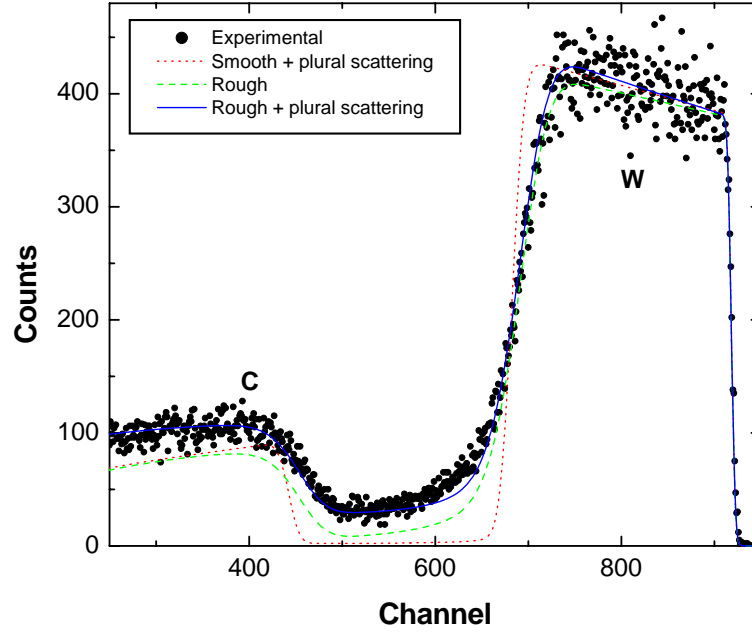


Figure 4.34.: 2.5 MeV protons backscattered from $3.5\ \mu\text{m}$ W on a rough carbon substrate, scattering angle 165° . Dots: Experimental data; Dotted line: Calculated spectrum for a smooth W layer ($3.6\ \mu\text{m}$) on a smooth C substrate including plural scattering; Dashed line: Calculated spectrum for a rough W layer ($3.5\ \mu\text{m}$, $\sigma = 0.30\ \mu\text{m}$) on a rough substrate (FWHM 20°); Solid line: As dashed line, but including plural scattering.

to the smooth layer the contribution of plural scattering has increased strongly, which is due to an enhancement of plural scattering at inclined incidence. The height and shape of the low energy tail below the W-peak in channels < 650 are determined by the wings of the tilt angle distribution with inclination angles $> \pm 45^\circ$. The observed tilt angle distribution, see Figure 4.32, is best described by a FWHM of 26.6° , while the best fit to the measured spectrum yields a FWHM of about 20° . It is assumed that inaccuracies in the measurement of the tilt angle distribution at high inclinations due to the apex angle of the profiler tip and the constant step width, together with uncertainties in the calculation of the plural scattering background, are the reason for this small discrepancy. Additionally it should be kept in mind that the used model of inclined line segments, see Figure 4.24(right), is only an approximation to physical reality, and the real surface has an additional fine structure.

The influence of the different roughnesses on the shape of the RBS spectrum is shown in more detail in Figure 4.35. The experimental data (black dots) and the solid line in the top and bottom figures are the same as in Figure 4.34. The substrate roughness is kept constant in Figure 4.35 (top), and the roughness of the W layer is varied from smooth to $0.6\ \mu\text{m}$. The roughness of the W-layer influences mainly the low energy edge of the W peak, best fit is obtained for $\sigma = 0.3\ \mu\text{m}$. The bottom part shows the influence of the carbon substrate roughness for constant W-layer roughness. Substrate roughness influences mainly the low energy tail below the W-peak, while the low energy edge of the W-peak is less affected by substrate roughness. Best fit is obtained

for about 20° FWHM. Due to the different effects of the two roughnesses on the shape of RBS spectra the two roughnesses can be easily distinguished.

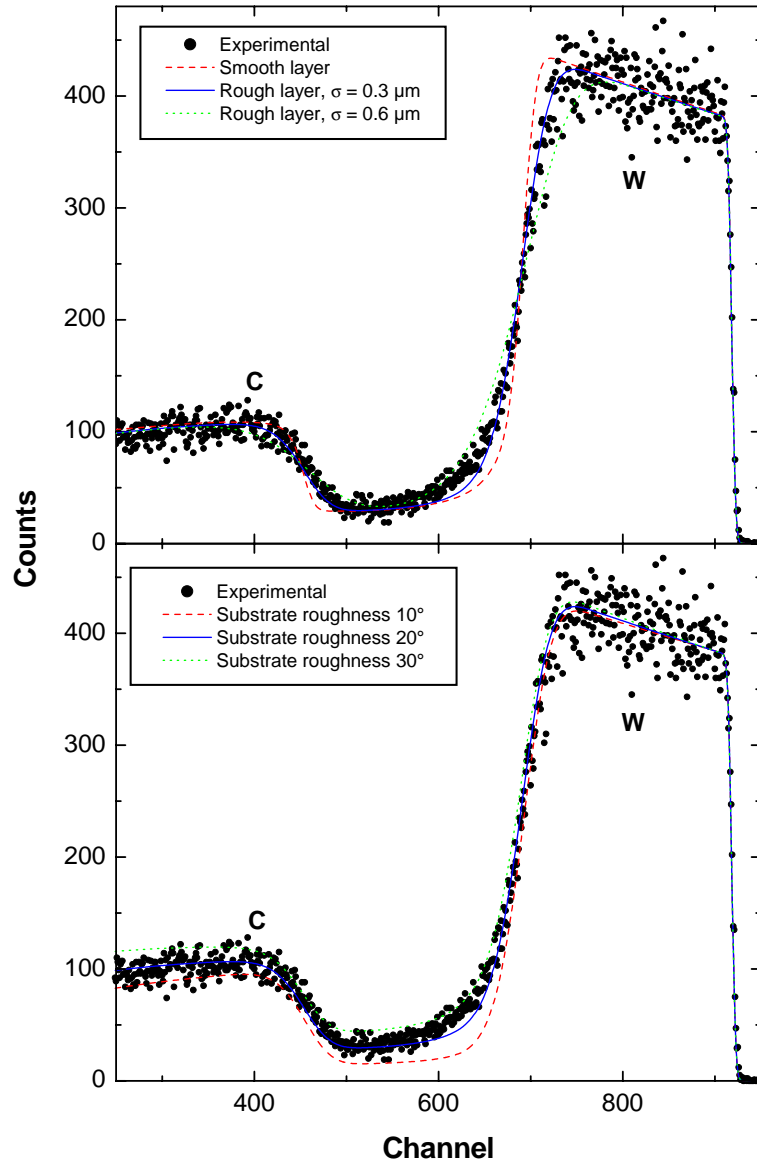


Figure 4.35.: Same experimental data as in Figure 4.34, compared to simulation calculations with different roughness parameters. Top: Calculations for a rough carbon substrate (FWHM 20°) and different W-layer roughnesses, characterized by a Gamma-distribution with standard deviation σ ; Bottom: Calculations for a rough W layer ($\sigma = 0.3 \mu\text{m}$) and different substrate roughnesses, characterized by a Lorentz-distribution of tilt angles with different FWHM's. Mean W-layer thickness $3.5 \mu\text{m}$, plural scattering included.

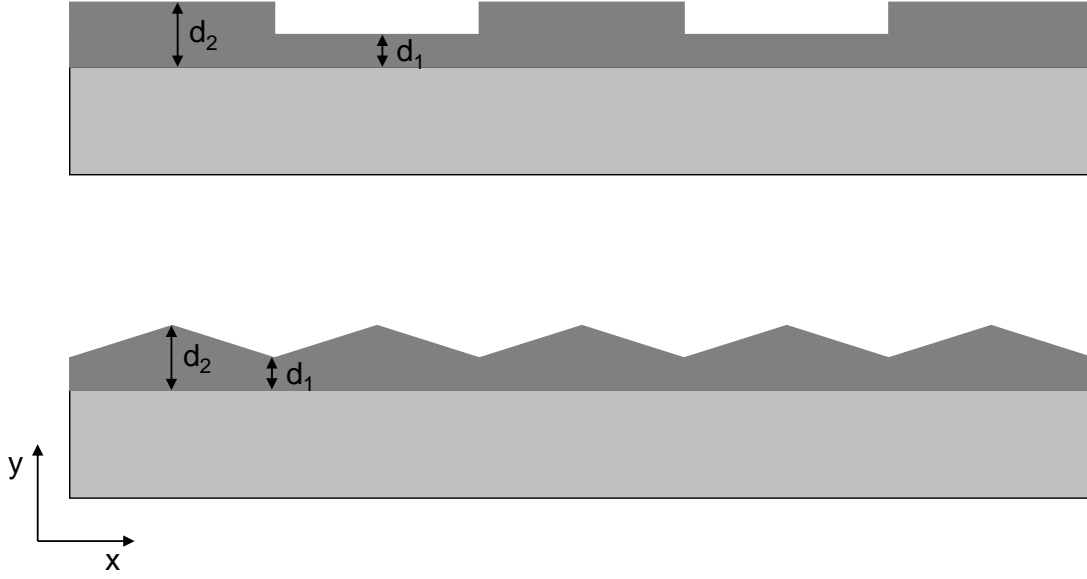


Figure 4.36.: Examples for lateral inhomogeneous layer structures, which can be simulated by SIMNRA.

4.11.3. Arbitrary lateral structures

The methods discussed in [subsection 4.11.1](#) are not limited to roughness in a strict sense, but can be used to calculate the effect of arbitrary lateral structures in the layer thickness distribution on IBA spectra. Some examples for possible structures are shown in [Figure 4.36](#).

As already pointed out in [subsection 4.11.1](#) the calculation is only approximate, because correlations between incident and exit beam (for example incidence through a valley and exit through a hilltop) are neglected. This approximation is only valid for large scattering angles close to 180° , and if the lateral extension of the lateral structures is much larger than the analyzed depth.

In order to simulate IBA spectra for an inhomogeneous lateral structure, the frequency (or probability) distribution of the layer thickness has to be supplied from an external file. See [subsubsection 3.6.1](#) and [Figure 3.15](#) for more information about the file format and how to use it.

The probability for an incident ion to hit a lateral position x is $q(x)$. For a homogeneous incident beam distribution this probability distribution is constant, i.e. $q(x) = c$. At each position x the layer thickness is $y = y(x)$ (see [Figure 4.36](#) for a definition of x and y) with the layer thickness distribution function $y(x)$. The frequency distribution $p(y)$ for the layer thickness y is then simply given by

$$p(y) dy = q(x) dx. \quad (4.99)$$

This gives immediately

$$p(y) = c \frac{dx}{dy}. \quad (4.100)$$

dx/dy is the derivative of the function $x(y)$, i.e. the inverse function of the layer thickness distribution function $y(x)$.

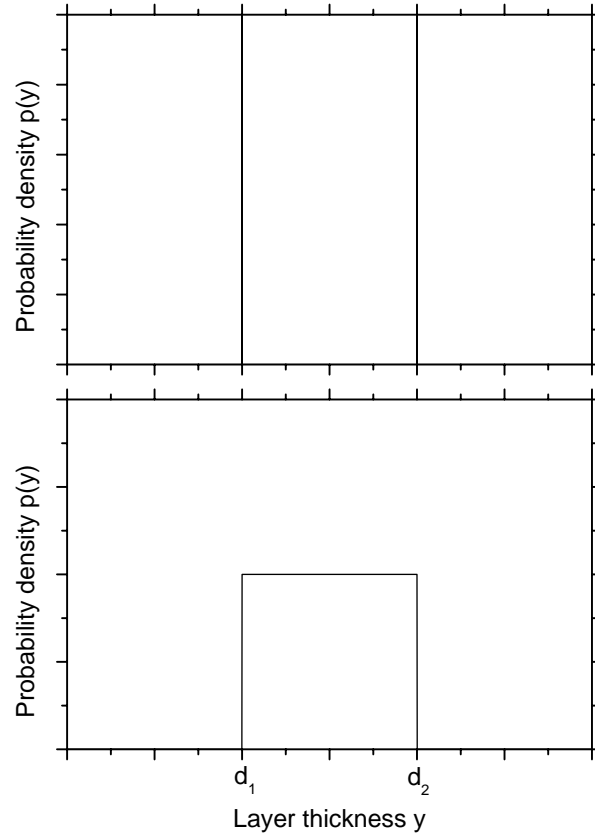


Figure 4.37.: Frequency distributions of the layer thickness y for the lateral structures shown in Figure 4.36. Top: Two δ -functions corresponding to the two layer thicknesses from Figure 4.36 top; Bottom: Frequency distribution for the structure from Figure 4.36 bottom.

The frequency distribution for a homogeneous layer thickness distribution with thickness d is the delta function $\delta(y - d)$. The layer thickness frequency distribution for the structure from Figure 4.36 top consists of two δ -functions, see Figure 4.37 top, the frequency distribution for the structure from Figure 4.36 bottom is constant from d_1 to d_2 , see Figure 4.37 bottom.

SIMNRA uses linear interpolation between the supplied data points for the frequency distribution before integration. SIMNRA normalizes the integral of the function to unity. The δ -function (for example at $y = 1000$ atoms/cm²) can be supplied in the following form:

0	0
999.9	0
1000	1
1000.1	0
2000	0

The integral of the above function would be 0.1.

N subspectra are used for the calculation of the total spectrum, where the number of subspectra N is given by the setting of the *Setup:Calculation:Roughness:Number of thickness steps*,

see [subsubsection 3.5.3](#). N has to be increased, if the spectrum contains visible steps.

4.12. Layer porosity

For laterally inhomogeneous samples IBA energy spectra yield information about the lateral distribution of material in the sample: This allows to determine the concentration and size of inhomogeneities (such as inclusions or pores) embedded in a different matrix material [3, 8, 14, 45, 53, 88, 106, 114].

The stochastic distribution of pores results in a fluctuation of the number of pores and of the path lengths inside of pores traversed by the ions on their incident and exit paths. These path-length fluctuations result in a fluctuation of energy losses on different ion trajectories: The presence of pores therefore results in an additional energy broadening, which can be measured and contains information about the size and density of the pores.

Stoquert/Szörényi [106] and Tosaki [114] assumed a Gaussian distribution for the number of pores traversed by a beam: If the beam traverses in mean N pores, then the standard deviation for the number of traversed pores is assumed to be \sqrt{N} . Together with some assumption about the shape of pores this relates the measured energy broadening with the concentration and size of pores. However, the assumption of a Gaussian distribution for the number of traversed pores is only valid at very low concentrations of pores in the matrix material and gets invalid at higher pore concentrations.

SIMNRA 6.60 and higher implements the following model of porous materials:

1. Pores are hollow, i.e. they are not filled with material and they have no stopping power or energy loss.
2. Pores are three-dimensional with spherical shape, see [Figure 4.38](#).
3. Pores are small, where small means that the change of the stopping power over the diameter of the pore (more precisely: the change of the stopping power in the surrounding matrix material) has to be sufficiently small. This is discussed in more detail in [subsection 4.12.2](#).
4. Pores are randomly distributed without short-range or long-range order.
5. Pores are allowed to overlap and can form networks of connected porosity, see [Figure 4.39](#).

4.12.1. The path length distribution function and its moments

The path length distribution inside pores can be calculated analytically if the concentration of pores is sufficiently small, i.e. if the probability that randomly placed pores overlap is sufficiently small. A random distribution of sparse pores is described by the Poisson distribution: If some volume contains in mean n pores, then the actual number of pores in a specific volume varies, and the number distribution in these volumes is described by the Poisson distribution. The Poisson distribution has variance n (or standard deviation \sqrt{n}) and converges towards a Gaussian distribution for larger n .

The unit cell is the volume which contains in mean one pore. We consider a cubical unit cell with side length L (see [Figure 4.38](#)). For spherical pores, see [Figure 4.38](#) with D the diameter

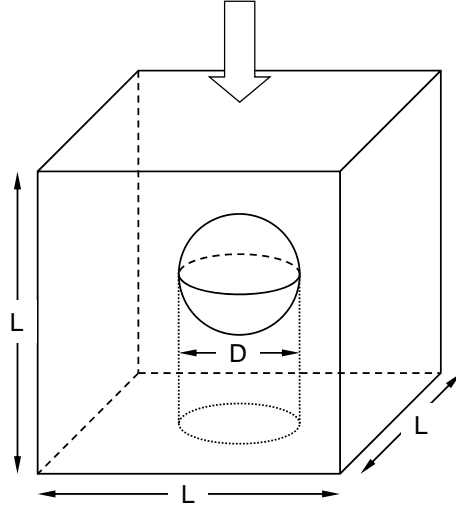


Figure 4.38.: Schematic representation of a spherical pore. D is the diameter of the pore, L is the length of the unit cell. The direction of incident ions is represented by the arrow.

and R the radius of pores, the volume fraction c_V of pores is given by

$$c_V = \frac{4\pi R^3}{3 L^3}. \quad (4.101)$$

The probability c , that an incident ion hits a pore, is given by

$$\begin{aligned} c &= \pi \frac{R^2}{L^2} \\ &= \left(\frac{9\pi}{16} \right)^{1/3} c_V^{2/3}. \end{aligned} \quad (4.102)$$

The first three moments of the path length distribution inside a single spherical pore are given by

$$\phi = \frac{2}{3}D \quad (4.103)$$

$$\nu = \frac{1}{18}D^2 \quad (4.104)$$

$$\mu = -\frac{1}{135}D^3, \quad (4.105)$$

with ϕ the mean value, ν the variance and μ the third central moment.

For Poisson-distributed spherical pores mean value F_I , variance V_I and third central moment M_I of the path length distribution inside of pores is given by:

$$F_I = \frac{2}{3}cD \quad (4.106)$$

$$V_I = \frac{1}{2}cD^2 \quad (4.107)$$

$$M_I = \frac{2}{5}cD^3. \quad (4.108)$$

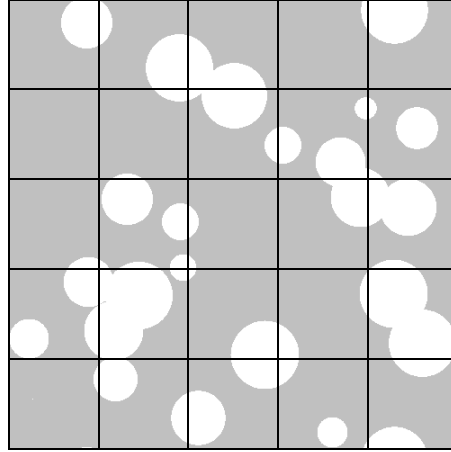


Figure 4.39.: Two-dimensional cross-section through the central plane of a typical Monte-Carlo distributions of spherical pores. The simulation volume consists of $5 \times 5 \times 5$ unit cells with periodic boundary conditions containing 125 pores, the volume fraction of pores is 20%.

The variance V_I is shown in Figure 4.40 (top), the central third moment M_I is shown in Figure 4.40 (bottom).

However, this analytical solution is only valid if the assumption of a Poisson distribution of pores is valid. This is generally only the case if pores are sparse, i.e. if their volume fraction is sufficiently small. For larger volume fractions the probability that pores overlap becomes non-negligible. Overlap of pores cannot be calculated analytically, but has to be treated by Monte-Carlo simulations of random arrangements of pores taking overlap into account.

Pores were randomly placed in a volume of $500 \times 500 \times 500$ pixels with periodic boundary conditions. A typical Monte-Carlo distribution with overlapping pores is shown in Figure 4.39 for a volume fraction of 20%. For simplicity only the 2-dimensional cross-section through the central plane of the simulation cube is shown. The volume fraction occupied by pores, the mean path length inside and outside of pores, the variance of the path length distribution and its third moment were calculated for each distribution.

The moments of the path length distributions inside pores, as obtained from the Monte-Carlo (MC) simulations, are shown in Figure 4.40. For volume fractions below about 5% V_I and M_I are well reproduced by the analytical solutions for non-interaction Poisson distributions, as presented in Equation 4.107 and Equation 4.108. At volume fractions above about 5% the MC results deviate from the Poisson distribution due to overlap of pores. The variances are generally smaller than expected from a Poisson distribution. For the practically important cases with volume fractions of 20–50% (these cases are often investigated experimentally), the difference is about 30–100%, so that a data evaluation with the assumption of a Poisson distribution results in substantial errors.

For thin porous layer, i.e. if the ion beam penetrates only 2–3 pores, the path length distribution function is non-Gaussian. For thicker porous layer (with penetration of 5 pores or more) the distribution function is close to Gaussian. The distribution function is usually slightly asymmetric due to the presence of a third moment (Figure 4.40(bottom)), but this asymmetry is

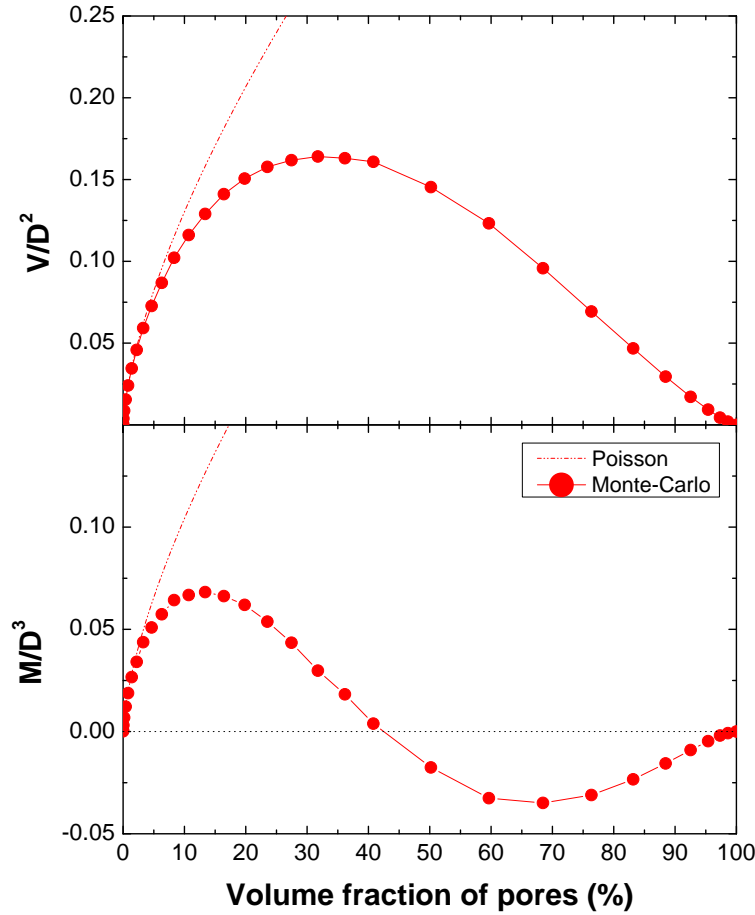


Figure 4.40.: Variance V (top), divided by D^2 , and third central moment M (bottom), divided by D^3 , of the path length distribution inside of pores after traversing one unit cell. D is the diameter of spherical pores, see Figure 4.38. Dashed lines: Poisson distribution of pores. Solid circles: Monte-Carlo simulation of overlapping spherical inclusions, see Figure 4.39.

typically small.

4.12.2. Relation between path length distribution and energy distribution

RBS, ERDA, and NRA measure energy distributions of particles coming from the sample. A distribution of path lengths in the material results in an energy distribution of the detected particles. The path length distribution function $p(x)$ for path lengths x in a material and the energy distribution function $q(E)$ for energies E are connected through $p(x)dx = q(E)dE$. The path length distribution function was already discussed in detail in [subsection 4.12.1](#). In this section the relations between the path length distribution and the corresponding energy distribution will be derived.

If a particle with initial energy E_0 traverses a path length x in a material, then its energy $E(x)$

4. Physics

is given by

$$E(x) = E_0 - \int_0^x S(E(x'), x') dx', \quad (4.109)$$

with the energy- (and generally depth-) dependent stopping power $S(E(x))$. In the fully general case, with arbitrarily large inclusion of any concentration, the relation between the path length distribution and the corresponding energy distribution can be derived only numerically through sampling of Monte-Carlo distributed inclusions. However, for sufficiently small inclusions the unit cell gets small enough, so that the stopping power can be approximated to be energy independent, i.e. the stopping power in the matrix is constant and S_M , and the stopping power in the inclusions is S_I . If a particle passes a path length x_M in the matrix and a path length x_I in inclusions, then its energy after passing a unit cell is given by

$$E(x_I, x_M) = E_0 - S_M x_M - S_I x_I. \quad (4.110)$$

E_0 is the particle energy at the entrance of the unit cell, and $x_I + x_M = L$. The mean energy after passing the unit cell $F_E = \int E q(E) dE$, the variance of the energy distribution $V_E = \int (E - F_E)^2 q(E) dE$, and the third central moment $M_E = \int (E - F_E)^3 q(E) dE$ are then given by

$$F_E = E_0 - S_I F_I - S_M F_M \quad (4.111)$$

$$V_E = (S_I - S_M)^2 V_I \quad (4.112)$$

$$M_E = -(S_I - S_M)^3 M_I. \quad (4.113)$$

F_I and F_M are the mean path lengths in inclusions and matrix, respectively, V_I the variance of the path length distribution in inclusions, and M_I its third moment. As already stressed above, Equation 4.111 to Equation 4.113 are only valid if the unit cell is sufficiently small, so that the stopping powers in the unit cell can be assumed to be constant. As stated above, this is the case for "sufficiently small inclusions at sufficiently high concentration": This statement requires some quantification.

In the general case, with $S_M \neq 0$ and $S_I \neq 0$, a quantification gets complicated. However, a simple estimate can be derived for hollow pores with $S_I = 0$. As already shown in [40], the stopping power can be expanded into a Taylor series at E_0 . By considering only the linear term of this expansion, the energy after passing a length x_M in the matrix can be approximated by

$$E(x_M) = E_0 - S x_M + \frac{1}{2} S S' x_M^2, \quad (4.114)$$

with the stopping power $S = S_M(E_0)$ at E_0 and its derivative $S' = dS_M/dE(E_0)$. By neglecting terms with S'^2 (which is justified if the stopping power changes only slowly with energy), the variance of the energy distribution can be calculated to be

$$V_E \approx S^2 (V_M - S' M_M - 2 S' F_M V_M). \quad (4.115)$$

The relative difference ϵ to the simpler Equation 4.112 is then given by

$$\epsilon = \left| S' \frac{M_M + 2 F_M V_M}{V_M} \right|, \quad (4.116)$$

i.e. Equation 4.112 is accurate within ϵ , if Equation 4.116 is smaller than ϵ . For spherical pores with a volume fraction of 10% this gives the relation $\epsilon \geq 1.19S'D$. With an accuracy for the variance of $\epsilon = 4\%$ (which means that the FWHM of the energy distribution is accurate within 2%) this gives for 3 MeV protons in carbon a maximum pore diameter of about $6.4 \mu\text{m}$, for 1.5 MeV ^4He ions in porous silicon the maximum pore diameter can be about 500 nm. At higher volume fractions the diameters can be even slightly smaller. In practice, the approximation of constant stopping powers in the unit cell is therefore a reasonable and sufficiently accurate approximation for many IBA applications.

Equation 4.111 to Equation 4.113 are valid for a single, sufficiently small unit cell as discussed above. But for thicker porous layers, where a number of unit cells is traversed by the ion beam, the stopping powers cannot be assumed to be energy-independent. However, in numerical simulations of spectra we still can approximate the stopping powers to be constant in each unit cell, provided the cell is sufficiently small. The energy dependence of the stopping powers then can be taken into account by evaluating the stopping powers $S_{M,i} = S_M(E_i)$ and $S_{I,i} = S_I(E_i)$, at the energy E_i , which is the energy at the entrance of unit cell i . For improved numerical accuracy E_i can be replaced by the mean energy in each unit cell $\bar{E}_i = E_i - 0.5F_{E,i}$, where $F_{E,i}$ is the mean energy loss in the unit cell according to Equation 4.111.

In addition to the energy dependence of the stopping power, which can be taken numerically into account by the calculational scheme outlined above, non-stochastic propagation of straggling due to the shape of the stopping power has to be taken into account for thicker targets. This is not only important within the porous layer itself, but also for the further propagation of straggling initially induced by porosity (for example if a porous layer is on top of a non-porous material). This effect is identical to the propagation of electronic energy-loss straggling, and can be calculated using the same methods already developed for the calculation of energy-loss straggling. A simple method for calculating the propagation of the straggling width is described for example in [109], a method for simultaneous propagation of width and third moment is shortly described in [78].

4.12.3. Combining in- and outgoing paths

IBA measurements are seldom performed in transmission geometry, but usually have a scattering or reaction event that connects an incident and an outgoing ion path. In most cases incident and outgoing paths are statistically independent, so that they can be combined by convolution and variances and third moments of inclusion-induced straggling on incident and outgoing paths can be added taking the kinematics of the scattering or reaction into account. Statistical independence requires a scattering or reaction angle sufficiently different from 180° : At 180° (and angles close to it) the same inclusions will be traversed by incident and outgoing beam, so that the path lengths get correlated and the distribution functions cannot be convoluted.

An example for an RBS spectrum from a porous layer is shown in Figure 4.41.

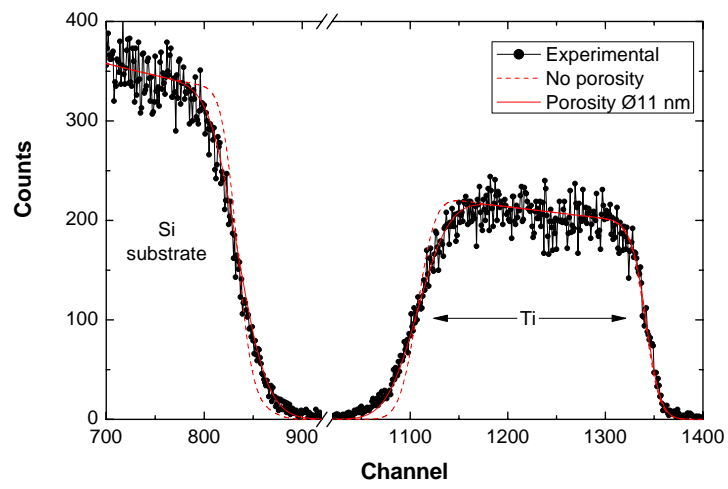


Figure 4.41.: RBS spectrum of a porous TiO_2 layer on Si, measured with 1 MeV ^4He . Dots: Experimental data; Dashed line: Computer simulation assuming a dense layer without porosity; solid line: Simulation assuming a porous layer with 11 nm pore diameter and 38% pore volume fraction. From [82].

4.13. Live time and pile-up

Nuclear spectroscopy systems are unavoidable sources of spectral distortions due to the finite pulse widths of the generated electronic signals and the paralyzable nature of the electronic system, which is not able to accept a second pulse while the previous is still being processed. Due to the finite pulse width there is always a probability of pulses to overlap, resulting in pulse pile-up, while the paralyzation of the system results in dead time losses of incident pulses. These distortions of the spectra can be minimized by decreasing the incident count rate (by decreasing the incident beam current or the detector solid angle), but they are usually clearly visible at higher count rates. These are often desirable due to the shorter measuring times.

This section describes how SIMNRA can take these effects into account in simulation calculations.

4.13.1. Live time correction

An Analog-Digital-Converter (ADC) requires a certain time to digitize an incident analog pulse. While the pulse is being processed, the ADC is not able to accept another pulse – this pulse is rejected. The time during which the electronics is busy (and unable to accept a pulse) is called *dead time* T_{Dead} . The *live time* T_{Live} is the time during which the electronics is able to accept pulses, while the *real time* of a measurement is called T_{Real} . These times are connected through the relation

$$T_{\text{Real}} = T_{\text{Live}} + T_{\text{Dead}}.$$

A pile-up rejector, which rejects overlapping pulses, is an additional source of system dead time.

The probability of a pulse to be detected is given by

$$P_{\text{Live}} = \frac{T_{\text{Live}}}{T_{\text{Real}}}, \quad (4.117)$$

because it is accepted only if it arrives during the system live time interval.

If a live-time correction is applied to simulated spectra (by checking *Apply live-time correction* in the *Live-time and pile-up* form, [subsubsection 3.5.2](#)), then the initial number of simulated counts n_k^0 in channel k is multiplied by the probability, that the counts actually are detected, to give the final number of simulated counts n_k :

$$n_k = P_{\text{Live}} n_k^0. \quad (4.118)$$

n_k^0 is the number of counts, which would be detected for an indefinitely small incident count rate (or an indefinitely fast ADC). Note that the live-time correction is applied to simulated spectra and *not* to experimental data – SIMNRA does not modify experimental spectra.

4.13.2. Calculation of pile-up

Due to the finite width of the electronic pulses there is always a probability of pulses to overlap. This phenomenon is called *pulse pile-up*. Pile-up can be minimized by decreasing the incident count rate (by decreasing the incident beam current or the detector solid angle), and by reducing

4. Physics

the width of the electronic pulses. Gaussian shaping with a peaking times of $0.5 \mu\text{s}$ is still long enough to preserve the optimal energy resolution of most semiconductor detectors used for RBS. A pile-up rejector can additionally decrease the pile-up level, which, however, can never achieve a complete elimination of pile-up effects.

SIMNRA offers two different models for calculating pile up. See [subsection 3.5.3](#) on how to select the different models, and for some remarks about computing times.

- *Accurate* model: This model is close to physical reality. Pulse shape effects and overlap of pulses according to their arrival-time distribution are taken into account correctly. The influence of a pile-up rejector circuit is modelled realistically. Major disadvantage of this model is its long computing time, especially for spectra with many channels.
- *Fast* model: This model is less accurate than the previous one, but can be calculated much faster.

The selection of the pile-up model depends on the use of a pile-up rejector circuit, the pulse shaping time, the desired level of accuracy, and the available computer power. In many cases, the *Fast* model already gives good results, especially if a pile-up rejector is used and the pulse rise time is larger than about $1 \mu\text{s}$. The *Accurate* model should be used only if necessary.

Accurate model

The effect of pulse pile-up is shown schematically in [Figure 4.42](#). Two overlapping pulses with heights i and j may be interpreted by the analyzing system as corresponding to a fictitious event with height k , and the ADC assigns to it an erroneous energy value. We will treat only double-pulse pile-up (i.e. the pile-up of exactly two pulses), while triple and other multiple pulse pile-up will be not considered – under realistic conditions for RBS or NRA other than double pulse pile-up is not observed [4].

The effect of pile-up on a single peak is shown schematically in [Figure 4.43](#). Two pulses arriving (almost) simultaneously result in a sum pulse with (almost) double height. Increasing time lag between the two pulses give sum pulses with heights between the original and the double height, resulting in a pile-up spectrum which extends from the original peak to the doubled channel number.

The mathematical models for pile-up calculations were already developed in the Seventies, and we adopt the model by Wielopolski and Gardner [125]. This model was initially developed for Gaussian shaped pulses, which are used by analog amplifiers. The model is also valid for modern Digital-Signal-Processors (DSP's), with the only difference that DSP's use digital filtering, corresponding to trapezoidal pulse shaping instead of analog Gaussian shaping.

Identifying pulse height with channel number (i.e. assuming linearity of the electronic system and no ADC offset), then the probability P_i of a pulse of height i in channel i is given by

$$P_i = \frac{n_i}{N},$$

where

$$N = \sum_{i=1}^m n_i,$$

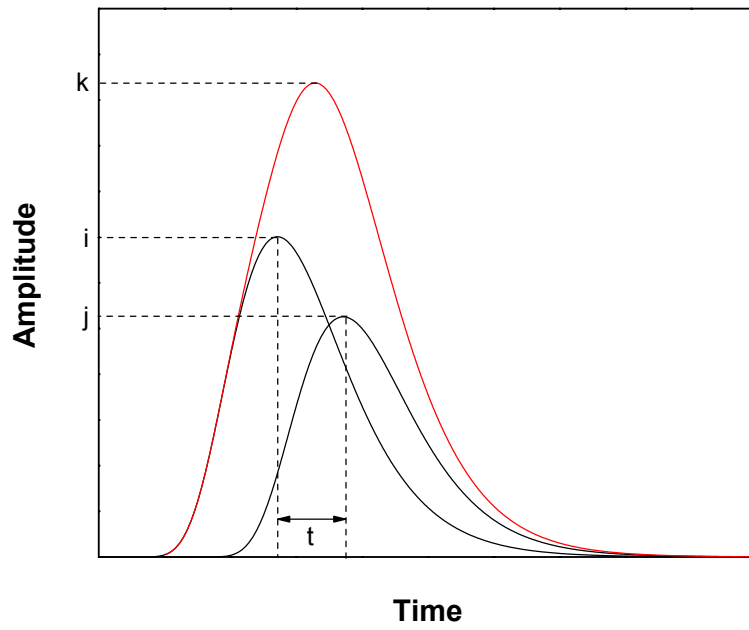


Figure 4.42.: Pile-up of a pulse with height i and a subsequent pulse with height j to a sum pulse of height k . t is the time interval between the two pulses.

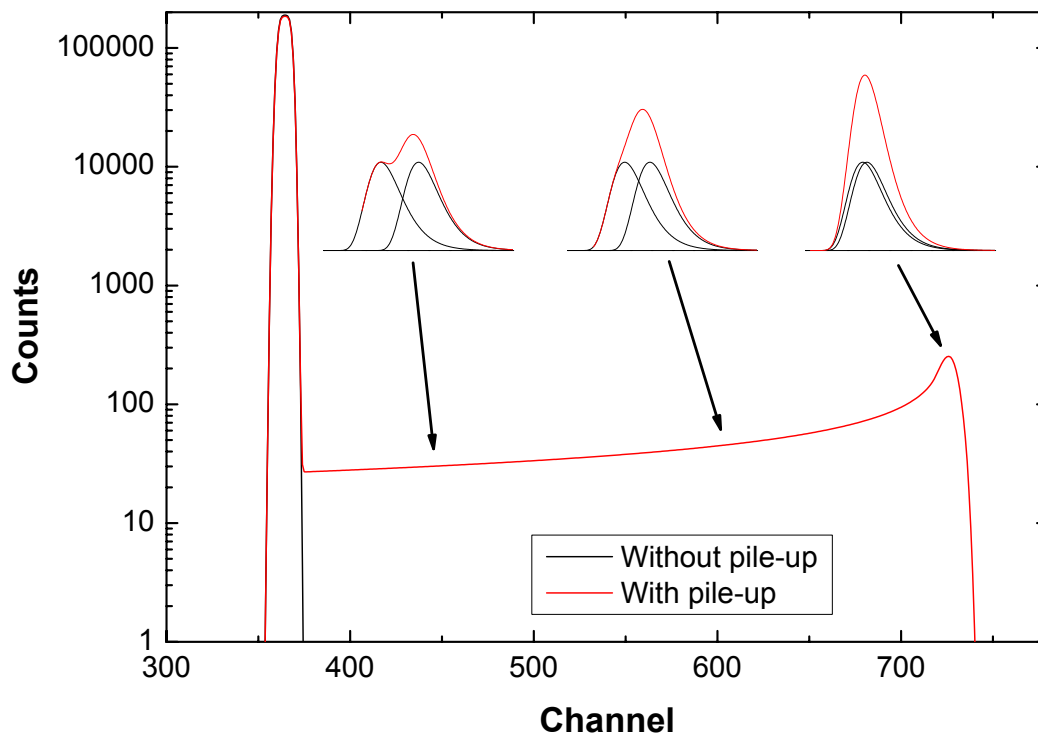


Figure 4.43.: Effect of pile-up on a single peak. The inset shows the individual and the sum pulses, which give the pile-up spectrum.

4. Physics

with n_i the number of count in channel i , m the total number of channels, and N the total number of counts in the spectrum. P_i is obtained from a simulated spectrum neglecting pile-up effects, or can be obtained from a low counting rate experiment in which the pile-up probability is negligible.

The time lag between any two pulses is distributed according to the interval probability distribution, which is derived from the Poisson probability distribution. The probability P_{ijk} is the probability, that a pulse of height i combines with a following pulse of height j to a pulse of height k , [Figure 4.42](#). In order to obtain a sum pulse of height k the second pulse j must arrive within a certain time interval $[t, t + dt]$. The probability P_{ijk} can be obtained as the product of the probabilities that

1. There is no pulse between time zero (when the initial pulse occurred) and time t :

$$P_1 = e^{-\alpha t}.$$

2. There is one pulse between time t and time $t + dt$.

$$P_2 = \alpha dt$$

3. There are zero pulses between time $t + dt$ and the end of the first pulse T_w .

$$P_3 = e^{-\alpha(T_w - t)}$$

For details see [\[125\]](#). The differential P_{ijk} is the product of these probabilities:

$$dP_{ijk} = \alpha e^{-\alpha T_w} dt$$

A pulse-height analyzer is only capable to analyze pulses into discrete channels (multi-channel analyzer). The P_{ijk} is obtained by integrating the differential probability over the time period that will give a pulse of size k . This time interval is from $t_1(i, j, k)$ to $t_2(i, j, k + 1)$, so that

$$P_{ijk} = \int_{t_1}^{t_2} \frac{dP_{ijk}}{dt} dt \quad (4.119)$$

$$= \alpha e^{-\alpha T_w} (t_2 - t_1) \quad (4.120)$$

$$= \alpha e^{-\alpha T_w} \Delta t_{ijk} \quad (4.121)$$

The probability P_{ijk} is proportional to the time increment Δt_{ijk} between the two pulses.

Unfortunately, the time increment Δt_{ijk} can be calculated analytically only for very simple pulse shapes. Wielopolski and Gardner [\[125\]](#) approximated the true pulse shapes by parabolic pulses, see [Figure 4.44](#) and [Figure 4.45](#). In that approximation, Δt_{ijk} is obtained from

$$\Delta t_{ijk} = T_p \left(\left[\frac{(i + j - k + 1)(i + j)}{ij} \right]^{1/2} - \left[\frac{(i + j - k)(i + j)}{ij} \right]^{1/2} \right), \text{ for } l \leq k \leq i + j, \quad (4.122)$$

where T_p is the pulse rise time (i.e. the time to reach the maximum value, see [Figure 3.10](#)), and l is whichever is largest of i and j . For parabolic pulses, we can use $T_w = T_p$. Other approximations (including the true pulse shape by numerical integration) can be found in [\[34\]](#).

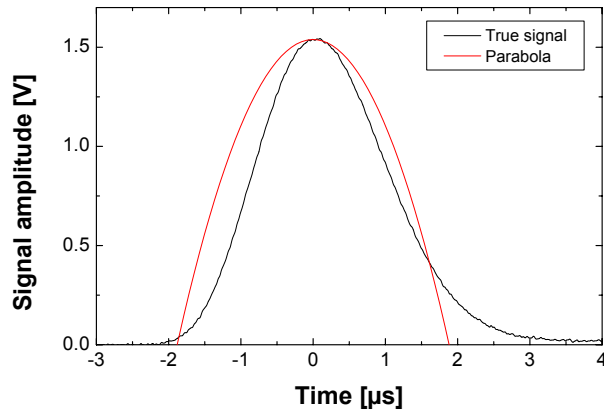


Figure 4.44.: Comparison of a real pulse (Ortec 672 spectroscopy amplifier, Gaussian shaping, shaping time $1 \mu s$) with the parabolic pulse approximation, pulse rise time $T_p = 1.9 \mu s$ according to Table 3.1.

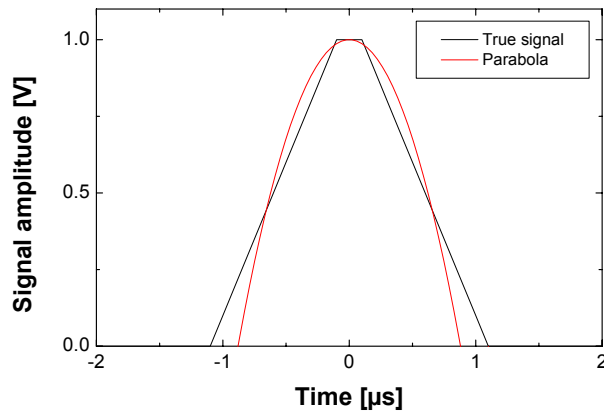


Figure 4.45.: Comparison of a trapezoidal pulse shape, as used by digital signal processors (DSP filter rise time $1 \mu s$, DSP flat top duration $0.2 \mu s$), with the parabolic pulse approximation, pulse rise time $T_p = 0.9 \mu s$ according to Table 3.1.

4. Physics

Pile-up results in losses and gains of pulses in each channel k . Losses in channel k are due to pile-up with pulses j into channel i , where the pulses k may pile up with a following pulse j , or may pile up with a preceding pulse j . The losses from channel k are given by

$$L_k = NP_k \sum_{j=1}^m P_j \sum_{i=a}^{j+k} (P_{kji} + P_{jki}), \quad (4.123)$$

where a is the largest of k and j .

The gain in channel k is due to pile-up of pulses i and j :

$$G_k = N \sum_{i=1}^{k-1} \sum_{j=p}^{k-1} P_i P_j P_{ijk}, \quad (4.124)$$

where $p = k - i$.

The content of each channel of the spectrum with pile-up n_k^{PU} is then given by

$$n_k^{PU} = n_k - L_k + G_k \quad (4.125)$$

$$= n_k - NP_k \sum_{j=1}^m P_j \sum_{i=a}^{j+k} (P_{kji} + P_{jki}) + N \sum_{i=1}^{k-1} \sum_{j=p}^{k-1} P_i P_j P_{ijk}, \quad (4.126)$$

with the channel content of the undistorted spectrum without pile-up n_k .

The total number of counts lost due to pile-up $L = \sum_{k=1}^m L_k$, and the total number of gained pulses $G = \sum_{k=1}^m G_k$ are connected through $L = 2G$, i.e. the total number of lost pulses is twice the number of gains.

Effect of a pile-up rejector A pile-up rejector (PUR) rejects pulse pairs, if their time lag t is larger than the pile-up rejector pair resolution time τ_2 . Pulses arriving almost simultaneously (i.e. with a time lag smaller than τ_2) are *not* rejected – A pile-up rejector therefore can only reduce the amount of pile-up, but is unable to eliminate it completely. The PUR pair resolution time can be found in the technical specifications of the spectroscopy amplifier and is usually $\leq 0.5 \mu s$, with typical values of 0.3 – $0.5 \mu s$.

A pile-up rejector is modelled in the following way: If the pile-up rejector is on, then pile-up occurs only, if the time lag t between the two pulses is smaller than τ_2 , i.e.

$$P_{ijk} = \begin{cases} \alpha e^{-\alpha T_w} (t_2 - t_1) & \text{if } t_1, t_2 \leq \tau_2 \\ 0 & \text{otherwise} \end{cases}$$

see Equation 4.120.

The effect of a pile-up rejector on pile-up from a single peak is shown in Figure 4.46. Pulse pairs arriving almost simultaneously are not influenced by the pile-up rejector, resulting in a pile-up peak at twice the channel number than the original peak. Pile-up of pulses with a time lag larger than the pair resolution time (which gives the pile-up signal between the two peaks, see Figure 4.43) is eliminated.

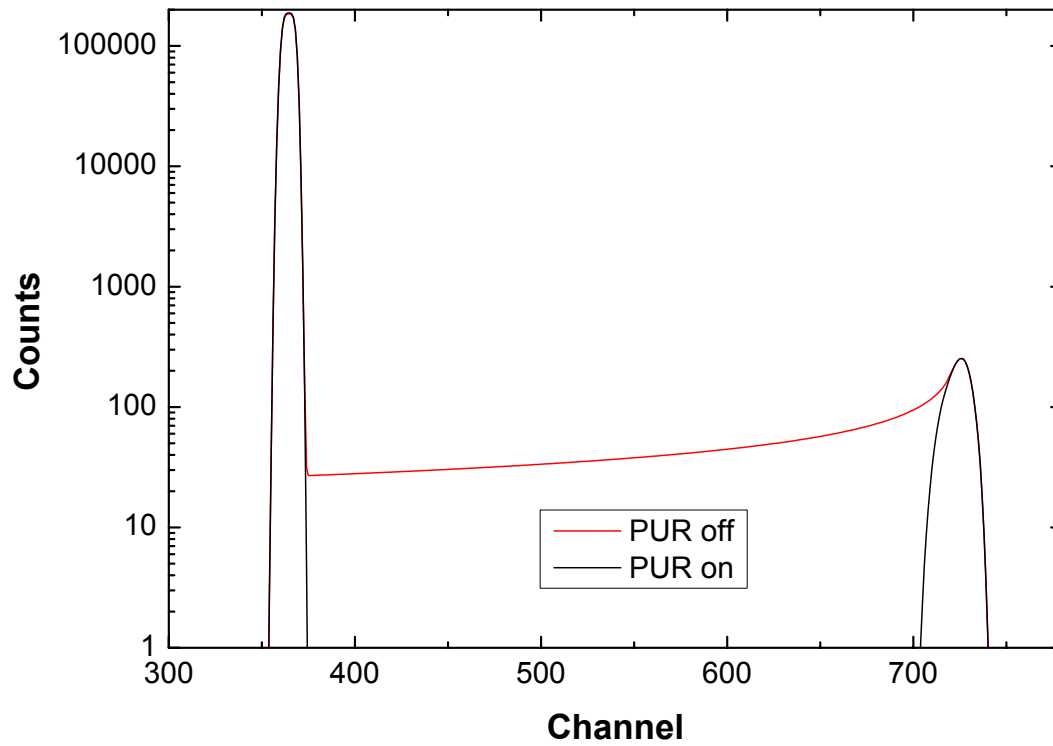


Figure 4.46.: Effect of a pile-up rejector (PUR) on pile-up from a single peak. The pile-up with a pile-up rejector switched off and on is shown. Pulse rise time $1\ \mu\text{s}$, pair resolution time $0.3\ \mu\text{s}$. The curve with PUR off is the same as in Figure 4.43.

4. Physics

Fast model

Major disadvantage of the *Accurate* model (subsubsection 4.13.2) is its long computing time. A much faster (but also less accurate) model for pile-up calculations was proposed by Jeynes [57]. In the Jeynes model (which is selected by the *Fast* pile-up model in SIMNRA) two pulses i and j always pile up to a pulse k with $k = i + j$. I.e., in the Jeynes model two pulses always arrive simultaneously, and

$$\Delta t_{ijk} = \begin{cases} T_f & \text{if } i + j = k \\ 0 & \text{otherwise} \end{cases}$$

where T_f is a fudge factor without physical meaning and the dimension of a time.

The losses L_k and gains G_k in each channel k then can be written as

$$L_k = 2ANn_k$$

$$G_k = A \sum_{i=1}^{k-1} n_i n_{k-i},$$

with a fudge factor A

$$A = \frac{T_f}{T_{\text{Real}}} e^{-\alpha T_f}.$$

The content of each channel of the spectrum with pile-up n_k^{PU} is then given by

$$n_k^{PU} = n_k - L_k + G_k \quad (4.127)$$

$$= n_k - 2ANn_k + A \sum_{i=1}^{k-1} n_i n_{k-i}, \quad (4.128)$$

with the channel content of the undistorted spectrum without pile-up n_k . The factor A (or T_f) can be freely adjusted to obtain best fit to a measured spectrum.

The *Accurate* and *Fast* models are compared in Figure 4.47. The *Fast* model cannot reproduce the pile-up contribution which occurs without a pile-up rejector, because it assumes that two pulses always arrive simultaneously, which is generally not the case. However, the *Fast* model is not too bad if a pile-up rejector is used and the pulse rise time is larger than the pair resolution time of the pile-up rejector: In this case, only pulses which arrive almost simultaneously are registered, and two pulses with heights i and j add up to a sum pulse of height $k \approx i + j$, as is assumed in the *Fast* model. The fudge parameter T_f then is identical to the pair resolution time τ_2 of the pile-up rejector.

The comparison of the *Accurate* and *Fast* models can be summarized as follows:

1. The *Fast* model can be applied, if a pile-up rejector is used and the pulse rise time T_p is larger than the pair resolution time τ_2 of the pile-up rejector. Because $\tau_2 \leq 0.5 \mu\text{s}$, this means that $T_p \geq 1 \mu\text{s}$.
2. The *Accurate* model has to be used, if the measurement was done without a pile-up rejector, or if short pulses are used, i.e. if the pulse rise time $T_p < 1 \mu\text{s}$.

The *Accurate* and *Fast* models are compared in Figure 4.48 to an experimental RBS spectrum. Both models describe the experimental pile-up quite well, although the *Accurate* model gives a slightly better approximation to the experimental spectrum.

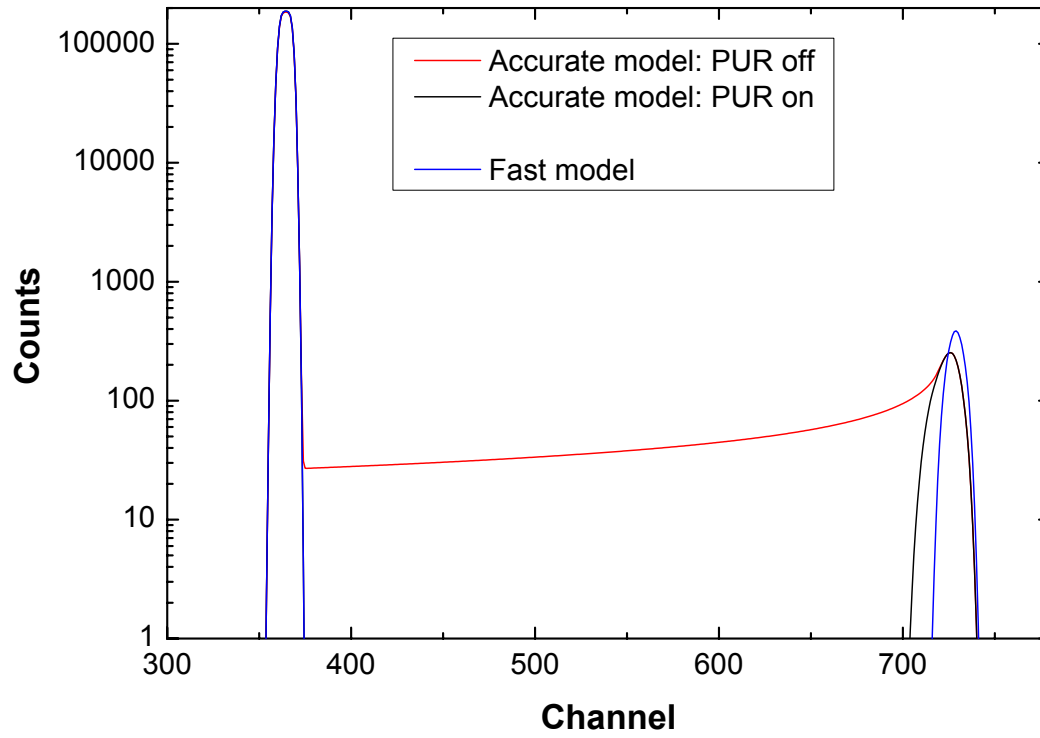


Figure 4.47.: Comparison of the *Accurate* and *Fast* models for pile-up from a single peak. *Accurate* model with pile-up rejector (PUR) on and off. Pulse rise time $1\ \mu\text{s}$, pair resolution time $0.3\ \mu\text{s}$, *Fast* model fudge time $0.3\ \mu\text{s}$.

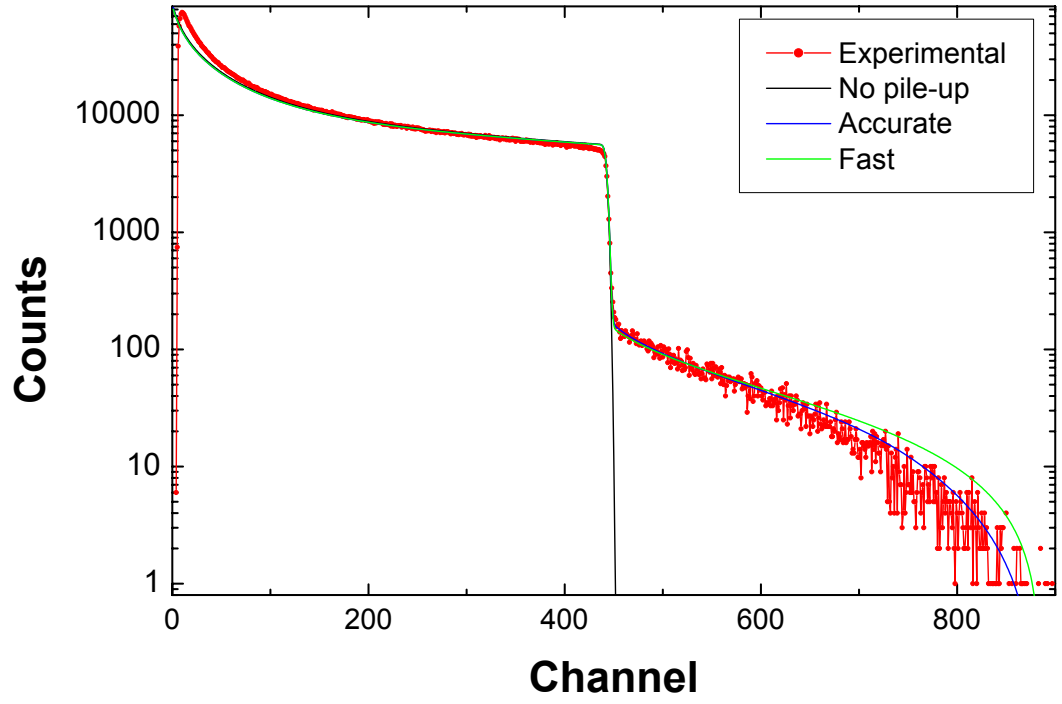


Figure 4.48.: 2 MeV ^4He backscattered from Au, $\theta = 165^\circ$. The experimental spectrum was measured with a Canberra 9660 DSP, filter rise time $0.5 \mu\text{s}$, filter flat top $0.1 \mu\text{s}$, pile-up rejector enabled, 6.5% dead time. Simulated spectra without pile-up; accurate model with $0.44 \mu\text{s}$ rise time (according to Table 3.1) and $0.35 \mu\text{s}$ PUR pair resolution time; fast model with $T_f = 0.32 \mu\text{s}$.

4.14. Expert system

SIMNRA provides a large number of different input data sets selectable by the user: There are six different stopping power models, three different straggling models, several models for the shape of straggling distributions, small and large angle scattering can be switched on or off, etc. This large number of possibilities renders it difficult to select the most accurate models for a given ion/target combination, ion energy and geometry - especially for inexperienced users. This problem cannot be overcome by appropriate default settings of simulation codes, because these are usually driven by programming or legal requirements, such as public availability of data or the permission to distribute additional third-party software: The default settings are therefore not necessarily the most accurate ones for a given setup, but the ones legally possible. Moreover, some settings (such as the dual scattering option in SIMNRA) increase accuracy, but at the cost of strongly increased computing time. The decision, if accuracy of simulations or a fast calculation is more important, cannot be made by software, but can be only made by the user. The selection of the most accurate settings requires experience and understanding of the underlying physics models. These can be gained by reading the literature, by reading the software documentation, by training courses, or by discussions with experienced colleagues working in the same field. But in any case, this process is time consuming, and even for experienced users oversights of clumsy parameter settings sometimes happens.

The SIMNRA expert system supports users in selecting the most accurate simulation settings for a given ion/target combination at a given energy and geometry. The expert system is a piece of artificial intelligence that emulates the ability and knowledge of a human IBA expert. It points out potential problems with the current simulation parameters and recommends model settings with enhanced accuracy for a given experiment in order to improve the quality and accuracy of IBA simulations and data analysis.

The expert system is implemented as system of If-Then rules emulating the knowledge of a human IBA expert. It points out potential problems with the current simulation setup through warnings and hints and recommends model settings with enhanced accuracy. The system output are recommendations. These can be observed, but without requirement. The final decision of parameter settings is always with the user, not the software.

The system has several levels of notifications:

- **Error:** An error is a setting that renders a calculation impossible. Usually these are formally incorrect parameter settings, for example incident ion energy below or equal to zero, no or impossible elements in the target, no selected cross-sections, etc. Error conditions must be solved before a simulation calculation can be performed.
- **Warning:** A warning is a setting that results in inaccuracies exceeding 2% over a larger energy range of the spectrum. Examples include the selection of an inaccurate stopping power model for the current ion/target combination, a difference of live and real times exceeding 2% while live time correction is switched off, etc.
- **Hint:** A hint is a setting that results in inaccuracies not exceeding 2% over a limited energy range of the spectrum. Typical examples are the selection of an inaccurate straggling

4. Physics

model, multiple and/or plural scattering are switched off for targets containing high-Z elements, etc.

- **Ok:** A setting that is considered to have best possible accuracy, or that results in inaccuracies not exceeding 0.1%.
- **Disregard:** A setting that has no influence on the simulation, typically because that parameter is not used with the current settings. A typical example would be a live time below 0 s (which is formally incorrect and otherwise would cause an *Error* condition) if live time correction is switched off: In that case that parameter does not play any role as it is not used in calculations.

Due to computing time limitations the level of inaccuracy usually cannot be calculated precisely, but has to be estimated based on simplified models. The distinction between *Hints* and *Warnings* is sometimes subjective and may reflect some prejudice of the author.

A typical output of the system is shown in [Figure 4.49](#). Every output consists of two parts: A description of the detected problem, and a recipe how to solve it. Underlined text in light blue are direct links that open the respective forms.

4.14.1. Performed tests

Formal correctness of simulation parameters

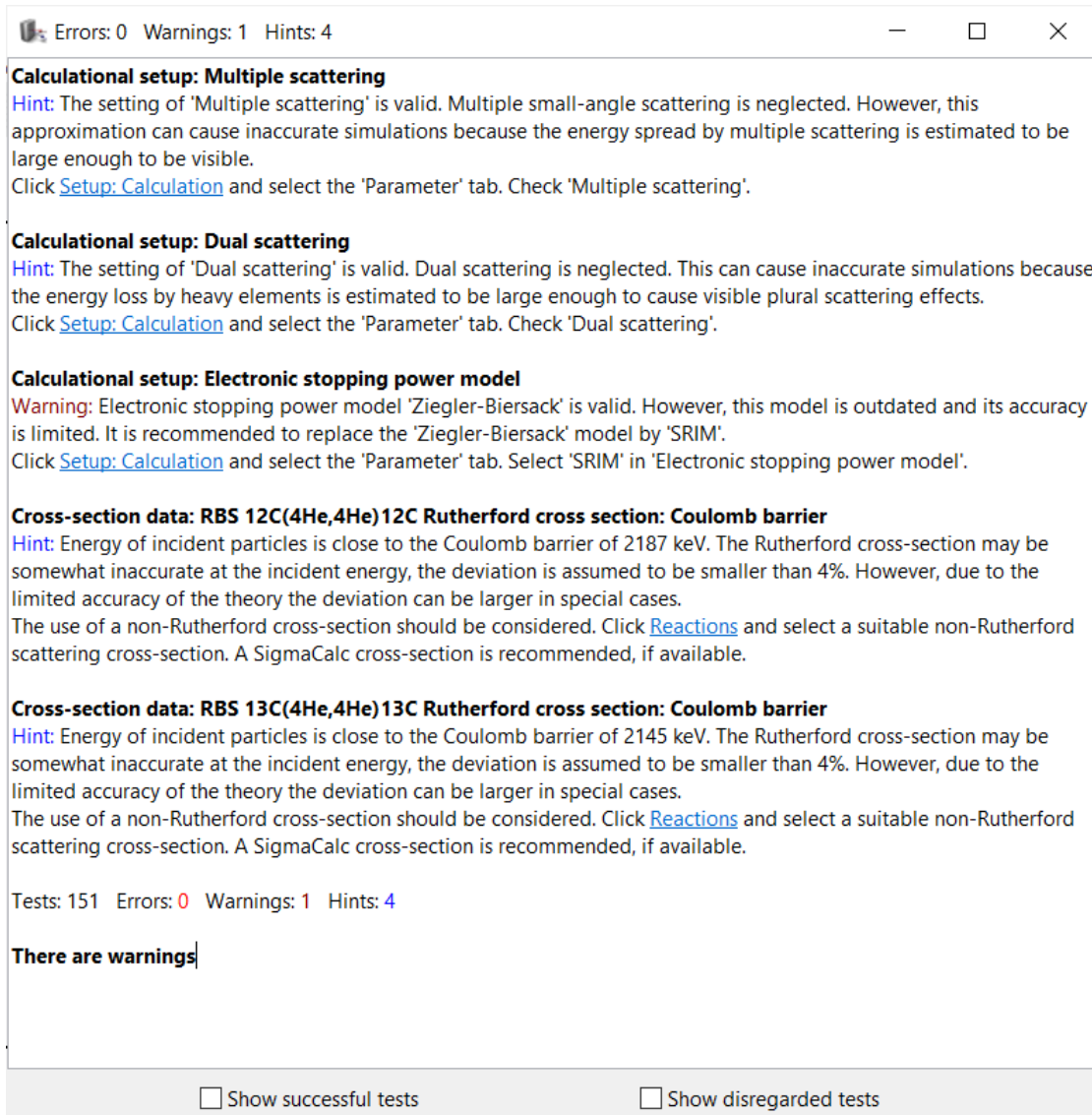
All parameters are checked for formal correctness, i.e., energies, times, lengths etc. must be larger (or larger equal where appropriate) than zero, the target must contain at least one element, all elements in the target must be valid, at least one cross-section must be selected, etc. Formal correctness of pairs of parameters is checked where appropriate, i.e., the real time must be larger or equal to the live time, etc.

Geometry of the experiment

Incident angle α , exit angle β and scattering angle θ must be compatible with a conic section at the sample surface, otherwise a *Warning* is issued.

Metadata of a measurement

An experimental IBA spectrum is always accompanied by metadata, such as date and time of measurement, name of the raw data file, parameters of the data acquisition system, etc. While some metadata are important for traceability of measurements and quality assurance, some other metadata are important for improving the accuracy of simulation calculations. These are real and live times of a measurement (necessary for live time correction and pile-up calculation), the shaping time of the amplifier system and information about the use of a pile-up rejector (necessary for pile-up calculation). These parameters are stored in some data file formats for experimental data, but may be lacking in others. If an experimental spectrum is present, then the corresponding metadata should be also present: Values for live and real time should be given, otherwise a *Hint* is issued. The live time correction should be switched on, otherwise a



Errors: 0 Warnings: 1 Hints: 4

Calculational setup: Multiple scattering
Hint: The setting of 'Multiple scattering' is valid. Multiple small-angle scattering is neglected. However, this approximation can cause inaccurate simulations because the energy spread by multiple scattering is estimated to be large enough to be visible.
 Click [Setup: Calculation](#) and select the 'Parameter' tab. Check 'Multiple scattering'.

Calculational setup: Dual scattering
Hint: The setting of 'Dual scattering' is valid. Dual scattering is neglected. This can cause inaccurate simulations because the energy loss by heavy elements is estimated to be large enough to cause visible plural scattering effects.
 Click [Setup: Calculation](#) and select the 'Parameter' tab. Check 'Dual scattering'.

Calculational setup: Electronic stopping power model
Warning: Electronic stopping power model 'Ziegler-Biersack' is valid. However, this model is outdated and its accuracy is limited. It is recommended to replace the 'Ziegler-Biersack' model by 'SRIM'.
 Click [Setup: Calculation](#) and select the 'Parameter' tab. Select 'SRIM' in 'Electronic stopping power model'.

Cross-section data: RBS 12C(4He,4He)12C Rutherford cross section: Coulomb barrier
Hint: Energy of incident particles is close to the Coulomb barrier of 2187 keV. The Rutherford cross-section may be somewhat inaccurate at the incident energy, the deviation is assumed to be smaller than 4%. However, due to the limited accuracy of the theory the deviation can be larger in special cases.
 The use of a non-Rutherford cross-section should be considered. Click [Reactions](#) and select a suitable non-Rutherford scattering cross-section. A SigmaCalc cross-section is recommended, if available.

Cross-section data: RBS 13C(4He,4He)13C Rutherford cross section: Coulomb barrier
Hint: Energy of incident particles is close to the Coulomb barrier of 2145 keV. The Rutherford cross-section may be somewhat inaccurate at the incident energy, the deviation is assumed to be smaller than 4%. However, due to the limited accuracy of the theory the deviation can be larger in special cases.
 The use of a non-Rutherford cross-section should be considered. Click [Reactions](#) and select a suitable non-Rutherford scattering cross-section. A SigmaCalc cross-section is recommended, if available.

Tests: 151 Errors: 0 Warnings: 1 Hints: 4

There are warnings

☐ Show successful tests ☐ Show disregarded tests

Figure 4.49.: Typical output of the expert system.

4. Physics

Warning or *Hint* is issued. The pile-up calculation should be switched on, otherwise a *Warning* or *Hint* is issued. Pure simulation calculations (without experimental spectrum) do not require metadata: In this case the lack of metadata is *Disregarded*.

Stopping power model

- SRIM 2003 [135] and later stopping powers are assumed to have good accuracy.
- Andersen-Ziegler (1977) [6, 131], Ziegler-Biersack (1985) [136], Konac et al. (1998) [61] stopping powers are assumed to be outdated, and a *Warning* is issued if these are used for simulation calculations.
- DPASS [94] stopping powers are assumed to have good accuracy for all ions heavier than He, and for all ions well below the stopping power-power maximum. DPASS stopping powers for H and He ions are assumed to have good accuracy at energies well above the stopping power maximum. However, a *Hint* is issued, that the spectrum may be less accurate at larger energy losses. DPASS stopping powers for H and He ions are assumed to be less accurate at energies in the vicinity of the stopping power maximum, and a *Warning* is issued if the incident energy is in this range.

Straggling model

- Bohr straggling [25] is assumed to have good accuracy only for incident H isotopes in lighter target elements at energies > 5000 keV/amu. In all other cases the accuracy is assumed to be limited and a *Hint* is issued.
- Chu straggling [35] neglects charge-state fluctuations and is assumed to have good accuracy only for incident H isotopes. In all other cases a *Hint* is issued.
- Yang straggling [128] is assumed to have good accuracy for all ion species.

Electronic screening to Rutherford cross-section

- The unscreened Rutherford cross-section [110] is assumed to be relatively inaccurate, and a *Warning* is issued.
- L'Ecuyer screening [66] is assumed to have good accuracy at backscattering angles $> 90^\circ$. At smaller angles a *Hint* is issued.
- Andersen [5] and Universal [136] screening are assumed to have good accuracy at all angles and all energies.

Coulomb barrier

The Coulomb barrier, i.e., the energy where the cross-section deviates by more than 4% from its Rutherford value, is calculated according to [122] for ^1H , ^2H , ^3H , ^3He , ^4He , ^6Li , ^7Li , ^9Be , ^{10}B , ^{11}B ,

^{12}C , ^{14}N , ^{16}O ions. A *Warning* is issued if the Rutherford cross-section is used at energies above the Coulomb barrier.

As discussed in [51], the theory from [122] has only limited accuracy. Therefore already at energies in the range 90% to 100% of the Coulomb barrier a *Hint* is issued.

Multiple and plural scattering

- If multiple small-angle scattering [7] is switched off and the energy loss by heavy elements (with $Z > 20$) exceeds 2.5% of the incident energy, then a *Hint* is issued.
- If plural scattering [42] is switched off and the energy loss by heavy elements (with $Z > 20$) exceeds 10% of the incident energy, then a *Hint* is issued.
- SIMNRA uses Rutherford cross-sections for the calculation of plural scattering. This may result in inaccuracies, if plural scattering is switched on and the energy of incident particles is above the Coulomb barrier, see above. In this case a *Warning* is issued.

Limitations of the expert system are its inability to detect incorrect experimental settings, such as incorrect incident energy or geometry. It cannot recommend cross-section data for non-Rutherford scattering or nuclear reactions, as in most cases the real accuracy of cross-section data sets is unknown. The expert system recommends SigmaCalc cross-sections [49] (if available for a given ion – target combination), as these are based on multiple measurements and a theoretical model and should have improved accuracy compared to individual measurements. The expert system also does not propose target structures based on a given experimental spectrum.

5. Examples

This chapter gives several examples for the abilities of SIMNRA. All backscattering spectra were measured at the IPP Garching at a scattering angle $\theta = 165^\circ$. The solid angle of the detector was 1.08×10^{-3} sr. A standard surface barrier detector with a nominal energy resolution of 15 keV FWHM was used.

5.1. RBS: Rutherford cross-sections

Figure 5.1 shows the measured and simulated spectra for 1.0 MeV ^4He incident ions on a gold layer with a thickness of about 100 nm on top of silicon. The simulated spectrum fits the measured data very well. The low background between the Si edge and the low energy Au edge is due to plural scattering (this means the backscattered particles have suffered more than one scattering event with large scattering angle) [19, 42, 104], which was not simulated for this example. The deviation between experiment and simulation at low energies in the Si spectrum is due to the same reason.

Figure 5.2 compares simulated spectra with single and dual scattering for 500 keV ^4He ions incident on a 100 nm gold layer on top of silicon with experimental data. At this low energy plural scattering is important. With the inclusion of dual scattering the experimental results are much better approximated. Dual scattering gives the background between the low energy edge of Au and the Si edge, and the steeper increase of the gold spectrum is better described. The results with dual scattering are slightly lower than the experimental results. This is due to trajectories with more than two scattering events, which are not calculated.

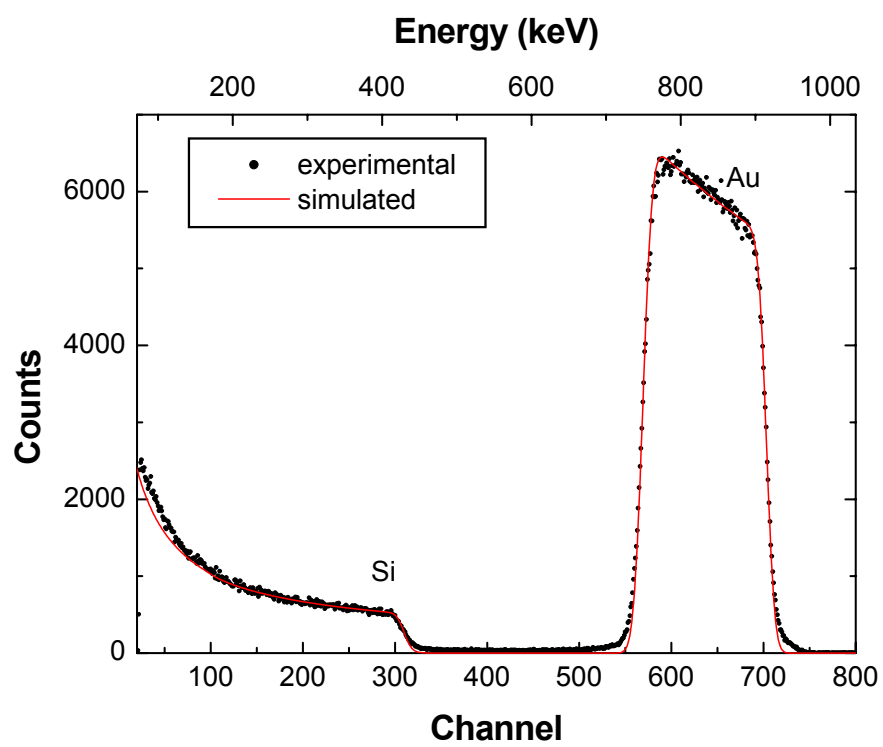


Figure 5.1.: 1000 keV ^4He incident on Au on top of silicon, $\theta = 165^\circ$.

5. Examples

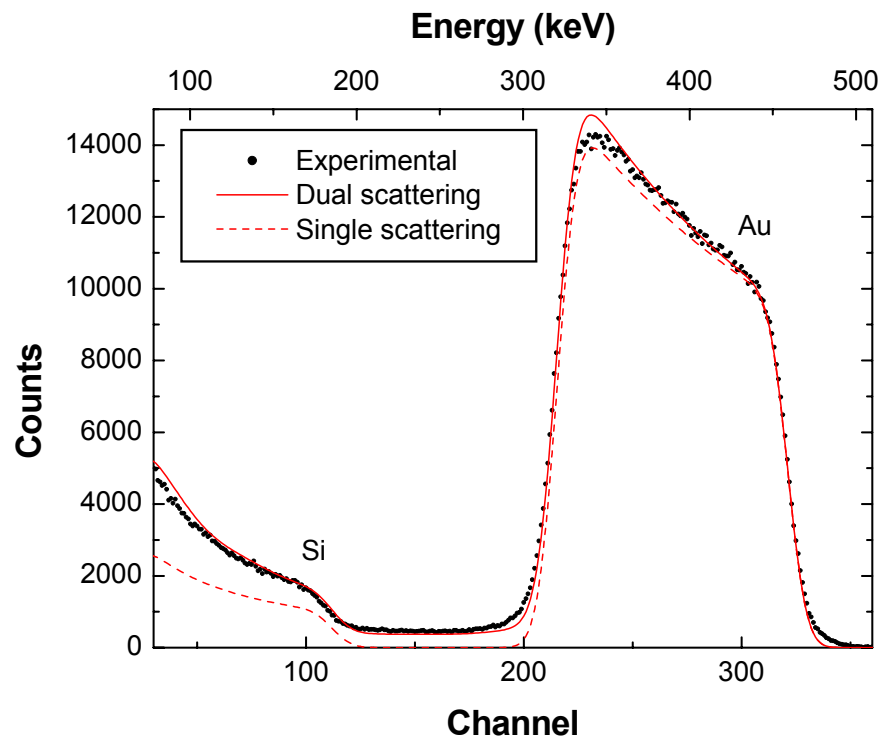


Figure 5.2.: 500 keV ^4He ions incident on 100 nm Au on top of Si, scattering angle 165° . Circles: experimental data points, dashed line: simulation with one scattering event, solid line: simulation with two scattering events.

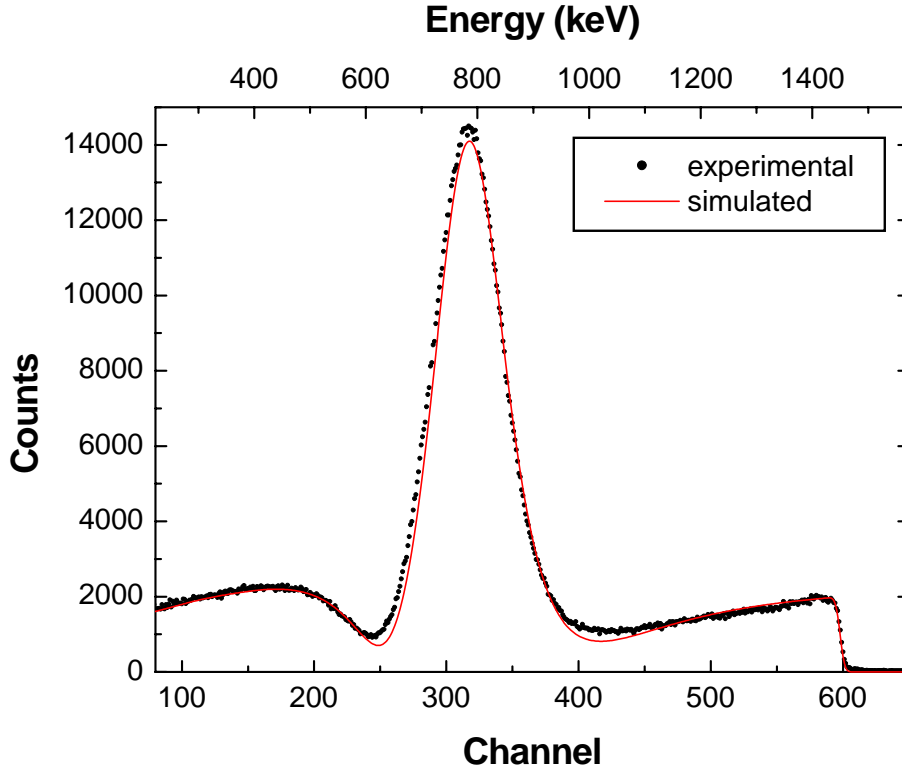


Figure 5.3.: 2000 keV protons on carbon (HOPG), $\alpha = 5^\circ$, $\theta = 165^\circ$.

5.2. RBS: Non-Rutherford cross-sections

Figure 5.3 shows the measured and simulated spectra for 2.0 MeV protons incident on highly oriented pyrolytic graphite (HOPG). To avoid channelling the incident angle α was 5° . The cross-section is non-Rutherford, and the cross-section data of Amirikas *et. al.* [1] were used for the simulation. The pronounced peak in the spectrum is due to the resonance in the $^{12}\text{C}(p,p)^{12}\text{C}$ cross-section at 1732 keV. The measured and simulated spectra agree very well.

Figure 5.4 shows the measured and simulated spectra for 2.0 MeV protons incident on silicon. To avoid channelling the incident angle α was 5° . The cross-section is non-Rutherford, and the cross-section data of Vorona *et. al.* [120] were used for the simulation. As in the case of carbon the measured and simulated spectra agree very well. The structures in the simulated spectrum between channel 500 and 700 are due to the experimentally determined cross-section data, which contain these structures.

5. Examples

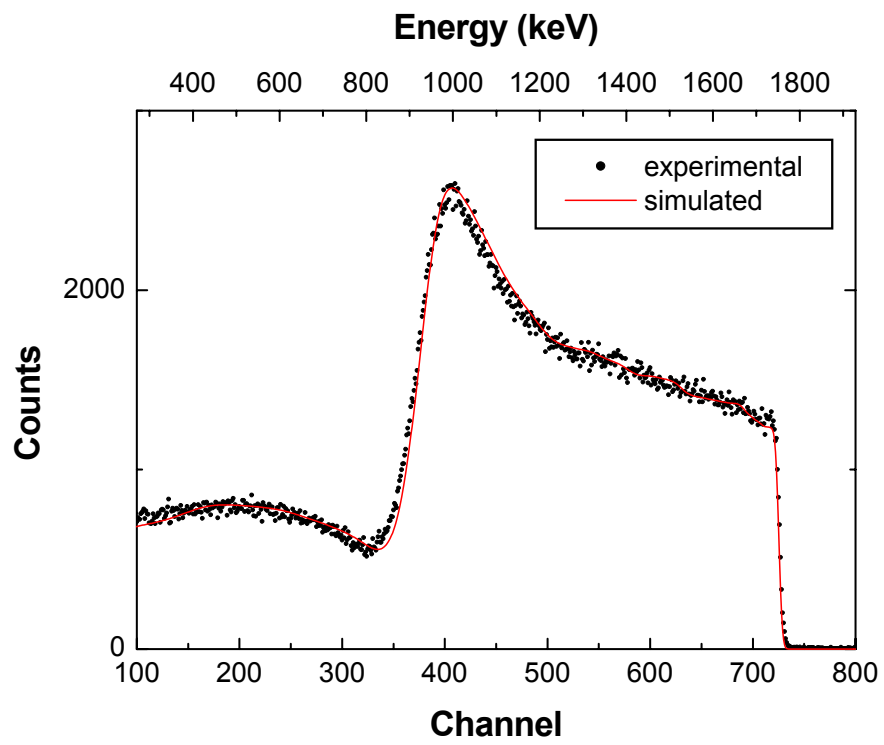


Figure 5.4.: 2000 keV protons backscattered from silicon, $\alpha = 5^\circ$, $\theta = 165^\circ$.

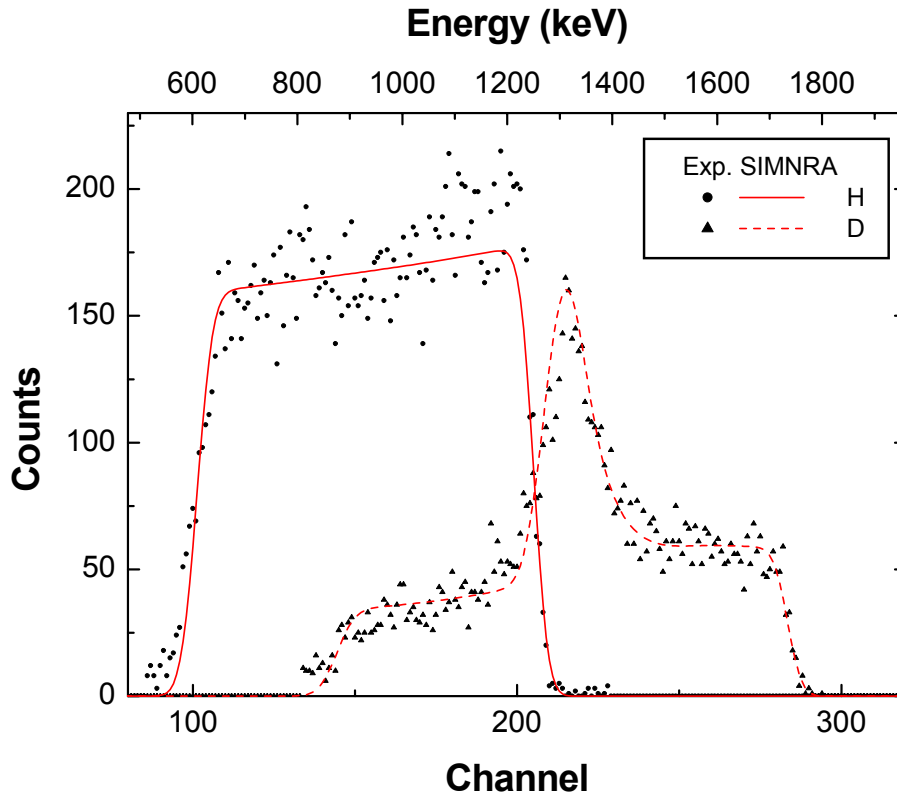


Figure 5.5.: ERDA with 2.6 MeV ^4He ions incident on a soft amorphous hydrocarbon layer (a:C-H layer) containing both H and D. The recoiling H and D atoms were separated with a ΔE -E telescope detector, the backscattered ^4He ions are not shown. $\alpha = \beta = 75^\circ$, $\theta = 30^\circ$.

5.3. ERDA: Non-Rutherford cross-sections

Figure 5.5 shows the measured and simulated spectra for ERDA with 2.6 MeV incident ^4He ions on a soft amorphous hydrocarbon layer (a:C-H layer) containing both H and D. The recoiling H and D atoms were separated with a ΔE -E telescope detector [126]. Both recoil cross-sections are non-Rutherford. The cross-section data of Baglin *et al.* [12] for $\text{H}(^4\text{He},\text{H})^4\text{He}$ and of Besenbacher *et al.* [21] for $\text{D}(^4\text{He},\text{D})^4\text{He}$ were used for the simulation. The peak in the deuterium spectrum is due to a resonance at a ^4He energy of 2130 keV. The measured and simulated data agree very well.

6. Acknowledgements

Most experimental cross-section data files are retrieved from the Ion Beam Analysis Nuclear Data Library (IBANDL) which is maintained and provided freely by the Nuclear Data Services of the IAEA (<http://www-nds.iaea.org>). IBANDL is a product of an IAEA Research Coordination Project and while all efforts were made to ensure that the most accurate information was adopted, the IAEA accepts no responsibility for the usage of IBANDL data.

The stopping power data and subroutines for Ziegler-Biersack stopping have been taken from Ziegler's SRIM 97 program, but were translated from BASIC to PASCAL.

The routines for reading RUMP's RBS file format were obtained from Peter Revesz, Cornell University, USA. They were translated from C to PASCAL.

The graphics subsystem for SIMNRA versions prior to 4.5 was developed by Achim von Keudell, Max-Planck-Institut für Plasmaphysik, Garching, Germany.

Code for fast calculations of the Gaussian and asymmetric Gaussian functions was obtained by Cleber L. Rodrigues, Universidade de São Paulo, Brazil.

SIMNRA uses libxml2 for reading and writing xml files.

Starting from version 5.84, SIMNRA uses Inno Setup for creating the setup program.

Valuable input and bug reports were obtained from Karl Ertl, Stefan Kapser, Karl Krieger, Thomas Köck, Hans Maier, Martin Oberkofler, Joachim Roth, and Barbara Wielunska (Max-Planck-Institut für Plasmaphysik, Garching, Germany), Jörg Röhrich and Swen Lindner (Hahn Meitner Institut, Berlin, Germany), Günther Dollinger (Technische Universität, München, Germany), Peter Revesz (Cornell University, USA), Andreas Gabrielsen (Norway), Caroline Raepsaet (Centre d'études de Saclay, France), Alexander Gurbich (Institute of Physics and Power Engineering, Obninsk, Russia), Jari Likonen (VTT, Espoo, Finland) and Tiago F. Silva (Universidade de São Paulo, Brazil).

Additional cross section data were obtained from Iva Bogdanović Radović (Rudjer Bošković Institute, Zagreb, Croatia), B. Diaz-Herrera (Max-Planck-Institut für Plasmaphysik, Garching, Germany), L.Y. Kim (Max-Planck-Institut für Plasmaphysik, Garching, Germany), Beata Tyburska (Max-Planck-Institut für Plasmaphysik, Garching, Germany), Herbert Kulinski (Max-Planck-Institut für Plasmaphysik, Garching, Germany), Massimo Chiari (I.N.F.N. - Sezione

di Firenze, Italy), Ana Rita Lopes Ramos (Nuclear and Technological Institute, Sacavém, Portugal), and Alexander Gurbich (Institute of Physics and Power Engineering, Obninsk, Russia).

The R33 file format and the R33Manager were developed by Ian Vickridge, Université Paris, France. The text of Appendix was written by I. Vickridge and has been taken from the file R33Help.htm.

The program *MottCalc* for the calculation of Mott scattering cross-sections has been developed at the National Technical University of Athens, Athens, Greece and was generously provided by M. Kokkoris.

The program *DPASS* for the calculation of DPASS stopping powers has been developed by A. Schinner, Johannes Kepler University, Linz, Austria, and P. Sigmund, University of Southern Denmark, Odense, Denmark. Its distribution with SIMNRA has been as generously permitted by P. Sigmund.

SIMNRA was developed at the Max-Planck-Institut für Plasmaphysik, Garching, Germany.

A. OLE automation reference

This chapter describes OLE 2.0 automation support in SIMNRA. SIMNRA is an OLE automation server, which allows other applications to control SIMNRA. All functions accessible through the user interface are also available through OLE functionality. This is useful for batch processing of a large number of spectra or for developing new applications. A short overview of the OLE objects and methods is given below, for a complete description of the parameters associated with OLE automation methods see the rest of [Appendix A](#). Some sample programs can be found in [section A.18](#).

A.1. Startup mode

SIMNRA starts hidden, if invoked through OLE automation¹. SIMNRA becomes visible with the command [App.Show](#).

A.2. Objects

SIMNRA exports the following OLE automation objects:

- [Simnra.App](#) — The application itself.
- [Simnra.Setup](#) — Experimental setup (excluding projectile).
- [Simnra.Projectile](#) — Projectile parameters.
- [Simnra.Calc](#) — Parameters for calculation.
- [Simnra.Target](#) — Target with all layers and elements.
- [Simnra.Foil](#) — Foil with all layers and elements.
- [Simnra.Window](#) — Beam window with all layers and elements.
- [Simnra.Fit](#) — Fit parameters.
- [Simnra.Spectrum](#) — Experimental and simulated spectra; Plot properties.
- [Simnra.PIGE](#) — Gamma-ray yields.
- [Simnra.Stopping](#) — Stopping powers, energy loss and straggling in elements and layers.

¹This behavior was introduced in version 6.79. Earlier versions started visible.

- *Simnra.CrossSec* — Information about available cross-section data sets, selects cross-section data sets for simulations, returns Rutherford cross-sections for scattering and recoils.

A.3. Properties and methods

SIMNRA exports the following OLE automation properties and methods, grouped by object:

A.3.1. Simnra.App

- *Active* — Specifies whether SIMNRA is active and has focus.
- *BorderStyle* — Sets the style of the SIMNRA main window.
- *BringToFront* — Brings SIMNRA to the front above all other applications.
- *BringToRear* — Brings SIMNRA behind all other applications.
- *CalculateDualScatteringBackground* — Calculates the dual scattering background.
- *CalculatePileup* — Applies a live-time correction and calculates pile-up for the simulated spectrum.
- *CalculateSpectrum* — Calculates a simulated spectrum.
- *CalculateSpectrumFast* — Calculates a fast simulated spectrum.
- *CalculateSpectrumToDepth* — Calculates a simulated spectrum until a specified depth.
- *CalculatingSpectrum* — Indicates if a spectrum is being calculated.
- *CloseButtonEnabled* — Specifies whether the *Close* button of the main form is enabled.
- *ControlsVisible* — Specifies if controls (main menu, toolbar, statusbar) are visible.
- *Copy* — Copies all data from one instance of SIMNRA to another.
- *CopySpectrumData* — Copies experimental and simulated spectra in ASCII format to the Windows clipboard.
- *CreateListOfCrSecs* — Creates the list of available cross-section data files.
- *DeleteSpectrumOnCalculate* — Specifies whether the current simulated spectrum is deleted if a new calculation is performed.
- *FileName* — Name of the SIMNRA data file.
- *FitSpectrum* — Fits a spectrum.
- *Height* — Height of the form, in pixels.

A. OLE automation reference

- *Hide* — Hides SIMNRA.
- *IncidentIonEnergyIsZero* — Specifies if the energy of the incident ion has decreased below the cutoff energy *Calc.EMin*.
- *LastMessage* — Text of the last error message or warning.
- *Left* — Left side of the form, relative to the screen in pixels.
- *LegendOutsideOfChart* — Specifies if the plot legend is drawn inside or outside of the chart.
- *LegendVisible* — Specifies if the plot legend is visible.
- *Maximize* — Maximizes SIMNRA to fill the whole screen.
- *MenuVisible* — Specifies if the main menu is visible.
- *Minimize* — Minimizes SIMNRA to the Windows task bar.
- *OLEUser* — Specifies which program controls the current SIMNRA instance.
- *Open* — Opens a XNRA, NRA or IDF-file with automatic determination of file type.
- *OpenAs* — Opens a XNRA, NRA or IDF-file with user-defined file type.
- *OpenStream* — Opens a XNRA or NRA file from a file stream.
- *Priority* — Sets the priority class of the application.
- *ReadSpectrumData* — Imports experimental data in different data formats.
- *Reset* — Resets the program to its starting values. All calculated spectra, target, foil and setup definitions are deleted.
- *Restore* — Restores the minimized application to its normal size.
- *SaveAs* — Saves a SIMNRA data file.
- *SaveThumbnailAs* — Saves a thumbnail of the current chart as bitmap file.
- *Show* — Shows SIMNRA, if it was hidden.
- *ShowMessages* — Specifies if error messages are shown.
- *SimulationMode* — Simulation mode: Either *NRA* for simulation of charged particle energy spectra or *PIGE* for simulation of Gamma-ray yields (PIGE), see [section 3.13](#).
- *SpectrumChanged* — Indicates if the calculated spectrum has changed, i.e. was re-calculated.
- *SRIMDirectory* — Specifies the directory of the SRIM program.

- *Standalone* — Specifies if SIMNRA is running standalone.
- *StatusbarVisible* — Specifies if the statusbar is visible.
- *ToolbarVisible* — Specifies if the SIMNRA toolbar is visible.
- *Top* — Top side of the form, relative to the screen in pixels.
- *TopAxisCaption* — Caption of the top axis.
- *Version* — SIMNRA version number.
- *Visible* — Indicates whether SIMNRA is visible.
- *Width* — Width of the form, in pixels.
- *WriteSpectrumData* — Writes all spectra (experimental, simulated) in ASCII-format to a file.

A.3.2. Simnra.Setup

- *Alpha* — Incident angle [deg].
- *ApertureDistance* — Distance between sample and detector aperture [mm].
- *ApertureHeight* — Height of the detector aperture [mm].
- *ApertureShape* — Shape of the detector aperture.
- *ApertureWidth* — Width of the detector aperture [mm].
- *BeamHeight* — Height of the incident beam [mm].
- *BeamShape* — Shape of the incident beam.
- *Beamspread* — Energy spread of the incident beam [keV FWHM].
- *BeamWidth* — Width of the incident beam [mm].
- *Beta* — Exit angle [deg].
- *CalibrationLinear* — Linear calibration term B for energy calibration, see Equation 3.1 [keV/channel].
- *CalibrationOffset* — Calibration offset A for energy calibration, see Equation 3.1 [keV].
- *CalibrationQuadratic* — Quadratic calibration term C for energy calibration, see Equation 3.1 [keV/channel²].
- *Copy* — Copies the experimental setup from one instance of SIMNRA to another.
- *DetectorPosition* — Detector position.

A. OLE automation reference

- *DetectorSensitivity* — Specifies if the detector sensitivity is unity or has been read from file.
- *DetectorSensitivityFileName* — Name of the detector sensitivity file.
- *DetectorResolution* — Detector resolution [keV FWHM].
- *DetectorType* — Type of detector (solid state, time-of-flight,...).
- *Energy* — Energy of incident ions [keV].
- *LiveTime* — Live time of a measurement [s].
- *LTCorrection* — Specifies if a live time correction is applied.
- *Open* — Reads the experimental setup from a XNRA, NRA or IDF-file with automatic determination of file type.
- *OpenAs* — Reads the experimental setup from a XNRA, NRA or IDF-file with user-defined file type.
- *ParticlesSr* — Number of incident particles times solid angle [sr].
- *PUCalculation* — Specifies if pile-up is calculated.
- *PUROn* — Pile-up rejector on or off.
- *PURResolution* — Pile-up rejector pair-resolution time [μ s].
- *ReadDetectorSensitivity* — Reads the detector sensitivity as function of energy from file.
- *RealTime* — Real time of a measurement [s].
- *RiseTime* — Rise time of the amplified pulse from zero to its maximum value [μ s].
- *SaveAs* — Saves the current experimental setup to a file.
- *SaveAsDefault* — Saves the current experimental setup as default.
- *SetBeta* — Sets exit angle β as function of the scattering geometry.
- *Theta* — Scattering angle [deg].
- *TOFTimeResolution* — Time resolution of a time-of-flight detector [ps].

A.3.3. Simnra.Projectile

- *Charge* — Nuclear charge of incident projectile.
- *Copy* — Copies the projectile from one instance of SIMNRA to another.
- *Mass* — Mass incident projectile [amu].
- *Name* — Name of incident projectile.

A.3.4. Simnra.Calc

- *Accuracy* — Accuracy of a calculation.
- *AutoStepwidthIn* — Specifies if automatic step width control for incident ions is used.
- *AutoStepwidthOut* — Specifies if automatic step width control for outgoing ions is used.
- *CalculateToEMin* — Specifies if a calculation is performed until the energy of incident ions decreases below the cutoff energy *EMin*.
- *Copy* — Copies the calculational setup from one instance of SIMNRA to another.
- *CreateSpectrum* — Specifies if a spectrum is calculated.
- *CreateSpectrumFromLayerNr* — Specifies that a spectrum is calculated from a specific layer.
- *CrossSecStraggling* — Specifies how structures in the cross-section and straggling are treated.
- *dEin* — Stepwidth incident ions [keV].
- *dEout* — Stepwidth outgoing ions [keV].
- *DualScattering* — Specifies if dual scattering is calculated.
- *DualScatteringRoughness* — Specifies how dual scattering and roughness are calculated simultaneously.
- *ElementSpectra* — Specifies if individual spectra for each element in the target are calculated.
- *EMin* — Cutoff energy [keV].
- *GeometricalStraggling* — Specifies if energy straggling by geometrical spread is calculated.
- *HighEnergyStopping* — Selects high energy stopping (Andersen-Ziegler only).
- *Isotopes* — Specifies if isotopes are taken into account.
- *IsotopeSpectra* — Specifies if individual spectra for each isotope in the target are calculated.
- *LogFile* — Specifies if a log file (SIMNRA.LOG) is created.
- *MultipleScattering* — Specifies if energy spread by small-angle multiple scattering is calculated.
- *MultipleScatteringModel* — Selects the multiple-scattering model.
- *NumberOfAngleVariations* — Number of angle steps in the calculation of rough substrates.

A. OLE automation reference

- *NumberOfDVariations* — Number of thickness steps in the calculation of rough layers.
- *NuclearStoppingModel* — Selects the nuclear energy-loss model.
- *Open* — Reads the calculational setup from a XNRA, NRA or IDF-file with automatic determination of file type.
- *OpenAs* — Reads the calculational setup from a XNRA, NRA or IDF-file with user-defined file type.
- *PUModel* — Selects the pile-up model.
- *SaveAs* — Saves the current calculational setup to a file.
- *SaveAsDefault* — Saves the current calculational setup as default.
- *ScreeningModel* — Selects the electronic screening model to the Rutherford cross section.
- *StoppingModel* — Selects the electronic energy-loss model.
- *Straggling* — Specifies if energy straggling is taken into account.
- *StragglingModel* — Selects the electronic energy-loss straggling model.
- *StragglingShape* — Selects the shape of straggling distributions.
- *SubstrateRoughnessDimension* — Dimensionality of substrate roughness.
- *ZBStopping* — Selects Ziegler-Biersack or Andersen-Ziegler stopping.

A.3.5. Simnra.Target

- *AddElement* — Adds an element to a layer.
- *AddElementProperties* — Adds multiple elements to a layer and sets the properties of these elements.
- *AddElements* — Adds multiple elements to a layer.
- *AddIsotope* — Adds an isotope to an element.
- *AddLayer* — Adds a layer to the target.
- *AddLayers* — Adds multiple layers to the target
- *Copy* — Copies the target from one instance of SIMNRA to another.
- *DeleteAllLayer* — Deletes all layers from the target.
- *DeleteElement* — Deletes an element from a layer.
- *DeleteIsotope* — Deletes an isotope from an element.

- *DeleteLayer* — Deletes a layer from the target.
- *ElementAmount* — Amount of an element in a layer [10^{15} atoms/cm²].
- *ElementConcentration* — Concentration of an element in a layer.
- *ElementConcentrationArray* — Concentrations of all elements in a layer.
- *ElementName* — Name of an element in a layer.
- *ElementZ* — Nuclear charge Z of an element in a layer.
- *ElementZArray* — Nuclear charges Z of all elements in a layer.
- *HasLayerPorosity* — Specifies if a layer is porous.
- *HasLayerRoughness* — Specifies if a layer is rough.
- *HasSubstrateRoughness* — Specifies if the substrate is rough.
- *InsertLayer* — Inserts a layer.
- *IsotopeConcentration* — Concentration of an isotope in an element.
- *IsotopeMass* — Mass of an isotope [amu].
- *LayerRoughness* — FWHM of the roughness of a layer [10^{15} atoms/cm²].
- *LayerThickness* — Thickness of a layer [10^{15} atoms/cm²].
- *NumberOfElements* — Number of different elements in a layer.
- *NumberOfIsotopes* — Number of different isotopes in an elements.
- *NumberOfLayers* — Total number of layers in the target.
- *PoreDiameter* — Diameter of pores in a layer [10^{15} atoms/cm²].
- *PorosityFraction* — Pore volume fraction in a layer.
- *ReadLayer* — Read a layer description from file.
- *ReadTarget* — Read a target description from file.
- *SaveLayerAs* — Save a layer description to file.
- *SaveTargetAs* — Save a target description to file.
- *StoppingFactor* — Correction factor to the stopping power of a layer, see [section 3.6.1](#).
- *SubstrateRoughness* — FWHM of the substrate roughness [deg].
- *SubstrateRoughnessDistribution* — Distribution function of the substrate roughness.
- *Thickness* — Total thickness of the target [10^{15} atoms/cm²].
- *TotalNumberOfElements* — Total number of different elements in the target.

A.3.6. Simnra.Foil

- *AddElement* — Adds an element to a layer.
- *AddLayer* — Adds a layer to the foil.
- *Copy* — Copies the foil from one instance of SIMNRA to another.
- *DeleteAllLayer* — Deletes all layers from the foil.
- *DeleteElement* — Deletes an element from a layer.
- *DeleteLayer* — Deletes a layer from the foil.
- *ElementAmount* — Amount of an element in a layer [10^{15} atoms/cm²].
- *ElementConcentration* — Concentration of an element in a layer.
- *ElementName* — Name of an element in a layer.
- *ElementZ* — Nuclear Charge Z of an element in a layer.
- *HasLayerPorosity* — Specifies if a layer is porous.
- *HasLayerRoughness* — Specifies if a layer is rough.
- *InsertLayer* — Inserts a layer.
- *LayerRoughness* — FWHM of the roughness of a layer [10^{15} atoms/cm²].
- *LayerThickness* — Thickness of a layer [10^{15} atoms/cm²].
- *NumberOfElements* — Number of different elements in a layer.
- *NumberOfLayers* — Total number of layers in the foil.
- *PoreDiameter* — Diameter of pores in a layer [10^{15} atoms/cm²].
- *PorosityFraction* — Pore volume fraction in a layer.
- *ReadLayer* — Read a layer description from file.
- *ReadFoil* — Read a foil description from file.
- *SaveLayerAs* — Save a layer description to file.
- *SaveFoilAs* — Save a foil description to file.
- *StoppingFactor* — Correction factor to the stopping power of a layer, see [section 3.6.1](#).
- *Thickness* — Total thickness of the foil [10^{15} atoms/cm²].

A.3.7. Simnra.Window

- *AddElement* — Adds an element to a layer.
- *AddLayer* — Adds a layer to the window.
- *Copy* — Copies the window from one instance of SIMNRA to another.
- *DeleteAllLayer* — Deletes all layers from the window.
- *DeleteElement* — Deletes an element from a layer.
- *DeleteLayer* — Deletes a layer from the window.
- *ElementAmount* — Amount of an element in a layer [10^{15} atoms/cm²].
- *ElementConcentration* — Concentration of an element in a layer.
- *ElementName* — Name of an element in a layer.
- *ElementZ* — Nuclear Charge Z of an element in a layer.
- *HasLayerPorosity* — Specifies if a layer is porous.
- *HasLayerRoughness* — Specifies if a layer is rough.
- *InsertLayer* — Inserts a layer.
- *LayerRoughness* — FWHM of the roughness of a layer [10^{15} atoms/cm²].
- *LayerThickness* — Thickness of a layer [10^{15} atoms/cm²].
- *NumberOfElements* — Number of different elements in a layer.
- *NumberOfLayers* — Total number of layers in the window.
- *PoreDiameter* — Diameter of pores in a layer [10^{15} atoms/cm²].
- *PorosityFraction* — Pore volume fraction in a layer.
- *ReadLayer* — Read a layer description from file.
- *ReadWindow* — Read a window description from file.
- *SaveLayerAs* — Save a layer description to file.
- *SaveWindowAs* — Save a window description to file.
- *StoppingFactor* — Correction factor to the stopping power of a layer, see [section 3.6.1](#).
- *Thickness* — Total thickness of the window [10^{15} atoms/cm²].

A. OLE automation reference

A.3.8. Simnra.Fit

- *Accuracy* — Desired accuracy of the fit.
- *Chi2* — Quadratic deviation χ^2 between the simulated and measured data points.
- *Chi2Evaluation* — Method of χ^2 evaluation.
- *EnergyCalibration* — Specifies if the energy calibration is fitted.
- *LayerComposition* — Specifies if the composition of a layer is fitted.
- *LayerNr* — Specifies which layer is fitted.
- *LayerRoughness* — Specifies if the roughness of a layer is fitted.
- *LayerThickness* — Specifies if the thickness of a layer is fitted.
- *MaxIterations* — Maximum number of fit iterations.
- *NumberOfRegions* — Number of different fit regions.
- *ParticlesSr* — Specifies if the number of incident particles times solid angle is fitted.
- *RegionMaxChannel* — Upper channels of fit regions.
- *RegionMinChannel* — Lower channels of fit regions.

A.3.9. Simnra.Spectrum

- *AddSimulated* — Adds all simulated spectra from one instance of SIMNRA to the simulated spectra of the current instance.
- *AutoScale* — Specifies if the plot is scaled automatically.
- *BottomAxisMax* — Bottom axis maximum.
- *BottomAxisMin* — Bottom axis minimum.
- *Copy* — Copies all spectra from one instance of SIMNRA to another.
- *CopyExperimental* — Copies the experimental spectrum from one instance of SIMNRA to another.
- *CopySimulated* — Copies all simulated spectra from one instance of SIMNRA to another.
- *Data* — Data in a specific channel of a spectrum.
- *DataArray* — Returns a spectrum.
- *Delete* — Deletes a spectrum.

- *HighElementID* — Returns the spectrum ID of the last element or isotope spectrum.
- *IDOfElement* — Returns the spectrum ID of a specific element.
- *IDOfIsotope* — Returns the spectrum ID of a specific isotope.
- *EndAcquisition* — Ending date and time of the acquisition of an experimental spectrum.
- *GetDataArray* — Returns a spectrum.
- *Integrate* — Integrates a spectrum.
- *LeftAxisMax* — Left axis maximum.
- *LeftAxisMin* — Left axis minimum.
- *LowElementID* — Returns the spectrum ID of the first element or isotope spectrum.
- *M* — Mass of the isotope of an isotope spectrum [amu].
- *NumberOfChannels* — Number of channels in a spectrum.
- *SetDataArray* — Sets a spectrum.
- *StartAcquisition* — Starting date and time of the acquisition of an experimental spectrum.
- *Title* — Title of a spectrum.
- *Z* — Nuclear charge of the element of an element spectrum.

A.3.10. Simnra.PIGE

- *Counts* — Number of counts in a Gamma-ray line.
- *E* — Energy of a Gamma-ray line [keV].
- *NumberOfGammas* — Number of Gamma-ray lines.

A.3.11. Simnra.Stopping

- *ClearCache* — Clears all internally cached stopping power data.
- *EnergylossInLayer* — Energy loss in a target, foil or window layer [keV].
- *StoppingInElement* — Stopping power in an element [keV/10¹⁵ atoms/cm²].
- *StoppingInLayer* — Stopping power in a target, foil or window layer [keV/10¹⁵ atoms/cm²].
- *StragglingInLayer* — FWHM of energy loss straggling in a target, foil or window layer [keV].

A.3.12. Simnra.CrossSec

- *Choose* — Selects a cross-section data set for a simulation calculation.
- *ClearCache* — Clears all internally cached cross-section data.
- *Count* — Returns the number of available cross-section data sets for all isotopes of all elements in the target.
- *EMax* — Maximum energy of a specific cross-section data set [keV].
- *EMin* — Minimum energy of a specific cross-section data set [keV].
- *ERDRutherford* — Returns the Rutherford cross-section for recoil creation, including screening by electrons [mb/sr].
- *FileName* — Returns the name of the data file of a cross-section data set.
- *Info* — Returns information about a cross-section data set.
- *ReactionsChosen* — Indicates if cross-section data are selected.
- *RBSRutherford* — Returns the Rutherford cross-section for scattering of the projectile, including screening by electrons [mb/sr].
- *SelectReactions* — Shows the reactions window for manual selection of cross-section data.
- *SelectRutherford* — Selects Rutherford cross-sections for one element.
- *SelectRutherfordAll* — Selects Rutherford cross-sections for all elements in the target.
- *SelectSigmaCalc* — Selects SigmaCalc cross-sections for one element.
- *SelectSigmaCalcAll* — Selects SigmaCalc cross-sections for all elements in the target.
- *Title* — Returns the title of a cross-section data set.
- *UniversalEMax* — Maximum energy for Universal screening function to Rutherford cross-section [keV].
- *Unselect* — Unselects all cross-sections for one element.
- *UnselectAll* — Unselects all cross-sections for all elements.
- *UnselectRutherford* — Unselects Rutherford cross-sections for one element.
- *UnselectRutherfordAll* — Unselects Rutherford cross-sections for all elements in the target.
- *UnselectSigmaCalc* — Unselects SigmaCalc cross-sections for one element.
- *UnselectSigmaCalcAll* — Unselects SigmaCalc cross-sections for all elements in the target.

A.4. Data types

SIMNRA is written in Borland Delphi and uses only automation compatible data types. The data types used by Delphi, the corresponding types in Microsoft's Interface Definition Language IDL, and the corresponding types used in Variants are summarized below.

Date/time values are stored in the following form: The integer part is the number of full days since 1899/12/30/ midnight, the fractional part is the fraction of the corresponding day.

Delphi type	IDL type	Variant type	Description
Boolean	VARIANT_BOOL	VT_BOOL	True = -1, False = 0
Double	double	VT_R8	8-byte real
Integer	long	VT_I4	4-byte signed integer
TDateTime	DATE		8-byte real
WideString	BSTR	VT_BSTR	Binary string of 16-bit wide characters

A.5. Simnra.App

The *Simnra.App* object represents the application itself.

A.5.1. Properties

Active

[Get] Property Active : Boolean;

Description

Specifies whether SIMNRA is active and has focus. *Active* is *true* while SIMNRA is active and *false* if it is not. SIMNRA is active if it has focus, and becomes inactive when a window from a different application is about to become activated. *Active* is readonly.

Related Properties and Methods

App.BringToFront [page 203](#)

App.Minimize [page 210](#)

App.Restore [page 215](#)

BorderStyle

[Get/Set] Property BorderStyle : Integer;

Default Value

A. OLE automation reference

0 (Sizeable)

Description

Specifies the style of the main program window of SIMNRA. Allowed values are:

- 0 Sizeable: The window can be sized and moved manually.
- 1 No border: The window has no visible border and cannot be sized or moved manually. It still can be sized or moved using the OLE functions [App.Width](#), [App.Height](#), [App.Left](#), [App.Top](#), [App.Minimize](#) and [App.Maximize](#).

Related Properties and Methods

App.ControlsVisible [page 193](#)

CalculatingSpectrum

[Get] Property CalculatingSpectrum : Boolean;

Description

Indicates if a spectrum is currently being calculated. *CalculatingSpectrum* is *true* while a spectrum is being calculated (by clicking *Calculate Spectrum*, *Calculate Spectrum Fast* or using one of the corresponding OLE functions), and *false* after the calculation has been finished. *CalculatingSpectrum* can be used for synchronizing SIMNRA with OLE clients. *CalculatingSpectrum* is readonly.

Related Properties and Methods

App.SpectrumChanged [page 200](#)

CloseButtonEnabled

[Set] Property CloseButtonEnabled : Boolean;

Description

Specifies whether the *Close* button of the main SIMNRA form (i.e. the red X in the upper right corner of the main form) is enabled. If *CloseButtonEnabled* is set to *false*, then the *Close* button is greyed out and SIMNRA cannot be closed by clicking this button. SIMNRA still can be closed by clicking [File: Exit](#) or ALT+F4. *CloseButtonEnabled* is writeonly.

Default Value

true

Related Properties and Methods

App.BorderStyle [page 191](#)
 App.ControlsVisible [page 193](#)
 App.MenuVisible [page 197](#)
 App.StatusbarVisible [page 201](#)
 App.ToolbarVisible [page 201](#)

ControlsVisible

[Set] Property ControlsVisible : Boolean;

Description

Setting *ControlsVisible* to *false* hides all controls (i.e. main menu, [toolbar](#) and [statusbar](#)) and leaves only the chart visible. Setting *ControlsVisible* is identical to setting [App.MenuVisible](#), [App.StatusbarVisible](#) and [App.ToolbarVisible](#) individually to the same value. *ControlsVisible* is write-only and can be only used to show or hide all controls. The current visibility of individual controls has to be retrieved from [App.MenuVisible](#), [App.StatusbarVisible](#) and [App.ToolbarVisible](#).

Related Properties and Methods

App.BorderStyle [page 191](#)
 App.MenuVisible [page 197](#)
 App.StatusbarVisible [page 201](#)
 App.ToolbarVisible [page 201](#)

DeleteSpectrumOnCalculate

[Get/Set] Property DeleteSpectrumOnCalculate : Boolean;

Default Value

true

Description

A. OLE automation reference

If *DeleteSpectrumOnCalculate* is *true*, the current simulated spectrum is deleted before a new spectrum is calculated. If *false*, the current simulated spectrum is conserved, and the new spectrum is added to the current one. Default is *true*.

Example:

```
' Spectrum for scattering angle 160°
```

```
Setup.Theta = 160
```

```
App.CalculateSpectrum
```

```
' Spectrum for scattering angle 170° is now added
```

```
' to the previous spectrum
```

```
App.DeleteSpectrumOnCalculate = false
```

```
Setup.Theta = 170
```

```
App.CalculateSpectrum
```

Related Properties and Methods

App.CalculateSpectrum [page 206](#)

FileName

[Get] Property FileName : WideString;

Description

Name of the currently used SIMNRA data file including full path. *FileName* is readonly. For changing *FileName* a SIMNRA data file has to be opened or saved.

Related Properties and Methods

App.Open [page 211](#)

App.SaveAs [page 215](#)

Height

[Get/Set] Property Height : Integer;

Description

Height of the main form, in pixels.

Related Properties and Methods

App.Left [page 195](#)
App.Top [page 202](#)
App.Width [page 203](#)

IncidentIonEnergyIsZero

[Get/Set] Property IncidentIonEnergyIsZero : Boolean;

Description

Specifies if the energy of the incident ion has decreased below the cutoff energy *Calc.EMin*. This property can be inspected after a call to *App.CalculateSpectrum* or *App.CalculateSpectrumFast*.

Related Properties and Methods

App.CalculateSpectrum [page 206](#)
App.CalculateSpectrumFast [page 206](#)
Calc.EMin [page 238](#)

LastMessage

[Get] Property LastMessage : WideString;

Description

Text of the last error message or warning. *LastMessage* is retained until a new error or warning is issued or it is read with *LastMessage*. *LastMessage* is readonly. See [section A.17](#) for more details about error handling.

Related Properties and Methods

App.ShowMessages [page 199](#)

Left

[Get/Set] Property Left : Integer;

Description

Position of the left side of the form, relative to the screen in pixels.

Related Properties and Methods

A. OLE automation reference

App.Height [page 194](#)

App.Top [page 202](#)

App.Width [page 203](#)

LegendOutsideOfChart

[Get/Set] Property LegendOutsideOfChart : Boolean;

Default Value

true

Description

If *LegendOutsideOfChart* is *true*, then the legend is drawn outside of the chart. If *LegendOutsideOfChart* is *false*, then the legend is drawn inside the chart. This maximizes the chart area, but may overlap the legend with spectra.

Related Properties and Methods

App.LegendVisible [page 196](#)

LegendVisible

[Get/Set] Property LegendVisible : Boolean;

Default Value

true

Description

If *LegendVisible* is *true*, then a plot legend is shown.

Related Properties and Methods

App.LegendOutsideOfChart [page 196](#)

MenuVisible

[Get/Set] Property MenuVisible : Boolean;

Default Value

true

Description

Indicates if the main menu is visible. Setting *MenuVisible* to *false* will hide the main menu.

Related Properties and Methods

App.ControlsVisible [page 193](#)

App.StatusbarVisible [page 201](#)

App.ToolbarVisible [page 201](#)

OLEUser

[Get/Set] Property OLEUser : WideString;

Default Value

”

Description

A. OLE automation reference

Calling *CreateOLEObject('Simnra.App')* will return a pointer to an already running instance of SIMNRA, if one or more instances of SIMNRA are already active: This may result in problems, if SIMNRA is controlled by two (or more) OLE clients simultaneously. *OLEUser* allows to determine if SIMNRA is already controlled by a user or by a different OLE client.

OLEUser returns

<i>User</i>	SIMNRA was invoked by a user
<i>ProgramName</i>	Name of the program controlling SIMNRA
"	The SIMNRA instance is not controlled by a user or by a different program

The following code creates a new instance of SIMNRA:

```
var
  SIMNRA : Variant;
begin
  repeat
    SIMNRA := CreateOLEObject(Simnra.App');
  until SIMNRA.OLEUser = "";
  SIMNRA.OLEUser := 'MyProgramName';
end;
```

It has to be noted that SIMNRA is not able to detect by itself if it is controlled by an OLE client or not. If your program controls SIMNRA then you should set *OLEUser* to the name of your program as shown in the code fragment above.

Related Properties and Methods

App.StandAlone [page 200](#)

Priority

[Get/Set] Property Priority : LongWord;

Default Value

NORMAL_PRIORITY_CLASS

Description

Priority class of the application, see the [Microsoft documentation of *SetPriorityClass*](#) for details.

Allowed values are:

ABOVE_NORMAL_PRIORITY_CLASS	0x00008000
BELOW_NORMAL_PRIORITY_CLASS	0x00004000
HIGH_PRIORITY_CLASS	0x00000080
IDLE_PRIORITY_CLASS	0x00000040
NORMAL_PRIORITY_CLASS	0x00000020

Note that *Priority* does not allow to set the priority class to REAL-TIME_PRIORITY_CLASS due to potentially negative effects on system stability.

ShowMessages

[Get/Set] Property ShowMessages : Boolean;

Default Value

false

Description

Specifies if error messages are shown or suppressed. If *ShowMessages* is *true*, program execution is stopped if an error is encountered, and a message box with an error description or warning is shown. Program execution is resumed after pressing the *OK* button. If *ShowMessages* is *false*, the message box is suppressed and program execution continues. The routine which produced the error, like [App.Open](#) or [App.CalculateSpectrum](#), returns an error flag. The text of the error message can be retrieved with [App.LastMessage](#). See [section A.17](#) for more details about error handling.

Related Properties and Methods

[App.LastMessage](#) [page 195](#)

SimulationMode

[Get/Set] Property SimulationMode : Integer;

Default Value

0

Description

Selects the simulation mode: Either *NRA* for simulation of charged particle energy spectra or *PIGE* for simulation of Gamma-ray yields (PIGE), see [section 3.13](#).

Possible values are:

0 : *NRA*

1 : *PIGE*.

SpectrumChanged

[Get/Set] Property SpectrumChanged : Boolean;

Description

Indicated if the calculated spectrum has changed due to a new calculation. SIMNRA will set *SpectrumChanged* to *true*, if a new spectrum has been calculated. But note that SIMNRA will never set *SpectrumChanged* to false – this has to be done by the OLE client. *SpectrumChanged* can be used to inform OLE clients that the calculated spectrum has changed. In that case the client should set *SpectrumChanged* to *false* after having obtained the spectrum.

Related Properties and Methods

App.CalculatingSpectrum [page 192](#)

SRIMDirectory

[Get/Set] Property SRIMDirectory : WideString;

Description

Returns the base directory of the SRIM program for stopping power calculations and allows to set this directory, see [section 3.11.2](#). The directory must be a fully qualified path including drive, for example C:\SRIM\. If *SRIMDirectory* is set, then this value is stored permanently in the registry.

Standalone

Function Standalone : Boolean;

Description

Indicated if SIMNRA is running standalone. It is running standalone if the program was invoked manually by a user. It is not standalone if the program was invoked by an OLE client.

StandAlone is outdated and should not be used for future developments. Use *App.OLEUser* instead.

Related Properties and Methods

App.OLEUser [page 197](#)

StatusbarVisible

[Get/Set] Property StatusbarVisible : Boolean;

Default Value

true

Description

Indicates if the [statusbar](#) is visible. Setting *StatusbarVisible* to *false* hides the statusbar.

Related Properties and Methods

App.ControlsVisible [page 193](#)

App.MenuVisible [page 197](#)

App.ToolbarVisible [page 201](#)

ToolbarVisible

[Get/Set] Property ToolbarVisible : Boolean;

Default Value

true

Description

Indicates if the [SIMNRA toolbar](#) is visible. Setting *ToolbarVisible* to *false* hides the toolbar.

Related Properties and Methods

A. OLE automation reference

App.ControlsVisible [page 193](#)

App.MenuVisible [page 197](#)

App.StatusbarVisible [page 201](#)

Top

[Get/Set] Property Top : Integer;

Description

Position of the top of the form, relative to the screen in pixels.

Related Properties and Methods

App.Height [page 194](#)

App.Left [page 195](#)

App.Width [page 203](#)

TopAxisCaption

[Get/Set] Property TopAxisCaption : WideString;

Default Value

'Energy (keV)'

Description

Caption of the top energy axis.

Version

[Get] Property Version : WideString;

Description

SIMNRA version number, as string. *Version* is readonly.

Visible

[Get] Property Visible : Boolean;

Description

Indicates whether SIMNRA is visible or not. The return value is *true* if SIMNRA is visible, and *false* otherwise.

Visible is readonly and can be used only to check the status of SIMNRA. It cannot be used to show or hide SIMNRA. To make SIMNRA visible call [App.Show](#). To make SIMNRA invisible call [App.Hide](#).

Related Properties and Methods

[App.Hide](#) [page 209](#)

[App.Show](#) [page 216](#)

Width

[Get/Set] Property Width : Integer;

Description

Width of the main form, in pixels.

Related Properties and Methods

[App.Height](#) [page 194](#)

[App.Left](#) [page 195](#)

[App.Top](#) [page 202](#)

A.5.2. *Methods*

BringToFront

Procedure BringToFront;

Description

Brings SIMNRA to the front above all other applications.

Parameters

None.

Return Value

None.

Related Properties and Methods

App.BringToRear [page 204](#)

App.Minimize [page 210](#)

BringToRear

Procedure BringToRear;

Description

Brings SIMNRA to the rear behind all other applications.

Parameters

None.

Return Value

None.

Related Properties and Methods

App.BringToFront [page 203](#)

App.Minimize [page 210](#)

CalculateDualScatteringBackground

Function CalculateDualScatteringBackground(AddToSpectrum : Boolean) : Boolean;

Description

Calculates the dual scattering background, i.e. the difference between the spectra with and without dual scattering, see [section 3.5.3](#). The dual scattering background spectrum is not visible in the graph, but can be accessed via the *spID* parameter with the methods of the *Simnra.Spectrum* object after *CalculateDualScatteringBackground* has been called.

The setting of *Calc.DualScattering* has no influence on *CalculateDualScatteringBackground*.

Usually there is no need to call *CalculateDualScatteringBackground*, because dual scattering is automatically calculated if *Calc.DualScattering* is *true* and a simulated spectrum is calculated by *App.CalculateSpectrum* or a related function. *CalculateDualScatteringBackground* can be used in special cases.

Parameters

AddToSpectrum Determines if the dual scattering background is added to the existing simulated spectrum. If *AddToSpectrum* is *true*, then the dual scattering background is added to the existing simulated spectrum, if *AddToSpectrum* is *false*, then the existing simulated spectrum is not modified.

Return Value

Returns *true* if the calculation succeeded.

Related Properties and Methods

Calc.DualScattering [page 237](#)

Simnra.Spectrum [page 188](#)

CalculatePileup

Function CalculatePileup : Boolean;

Description

Applies a live-time correction and calculates pile-up for the simulated spectrum, see [subsubsection 3.5.2](#). [Setup.LTCorrection](#) must be *true* for a live-time correction and [Setup.PUCalculation](#) must be *true* for a pile-up simulation; [Setup.LiveTime](#) and [Setup.RealTime](#) must be greater than zero. Note that *CalculatePileup* will do nothing if [Setup.LTCorrection](#) and [Setup.PUCalculation](#) are both set to *false*.

Under normal circumstances *CalculatePileup* should not be used, because the live time correction and pile-up simulation are performed automatically if [App.CalculateSpectrum](#) (or a related function) is used. *CalculatePileup* should be used only, if previous simulations were performed without live time correction and pile-up simulation. In these cases a live time correction and pile-up simulation can be applied at the very end by using *CalculatePileup*. This may be due to performance reasons or for superimposed spectra.

Parameters

None.

Return Value

Returns *true* if the calculation succeeded.

Related Properties and Methods

A. OLE automation reference

App.CalculateSpectrum [page 206](#)

Setup.LiveTime [page 224](#)

Setup.LTCorrection [page 224](#)

Setup.PUCalculation [page 225](#)

Setup.RealTime [page 226](#)

CalculateSpectrum

Function CalculateSpectrum : Boolean;

Description

Calculates a simulated spectrum. See [section 3.8](#) for details and differences to [App.CalculateSpectrumFast](#).

Parameters

None.

Return Value

Returns *true* if the calculation succeeded.

Related Properties and Methods

App.CalculateSpectrumToDepth [page 207](#)

App.CalculateSpectrumFast [page 206](#)

App.CalculatingSpectrum [page 192](#)

App.DeleteSpectrumOnCalculate [page 193](#)

CalculateSpectrumFast

Function CalculateSpectrumFast : Boolean;

Description

Calculates a fast simulated spectrum. See [section 3.8](#) for details and differences to [App.CalculateSpectrum](#).

Parameters

None.

Return Value

Returns *true* if the calculation succeeded.

Related Properties and Methods

[App.CalculateSpectrum page 206](#)
[App.CalculateSpectrumToDepth page 207](#)
[App.CalculatingSpectrum page 192](#)
[App.DeleteSpectrumOnCalculate page 193](#)

CalculateSpectrumToDepth

Function CalculateSpectrumToDepth(Depth : Double) : Boolean;

Description

Calculates a simulated spectrum until a specified depth. The calculation is finished when the specified depth is reached. *CalculateSpectrumToDepth* is identical to *CalculateSpectrum*, but it is considerably faster for small values of *Depth*. *CalculateSpectrumToDepth* should not be used for spectrum simulation, but can be used for depth-resolution calculations and the like.

Parameters

<i>Depth</i>	Maximum depth of calculation [10^{15} atoms/cm ²].
--------------	---

Return Value

Returns *true* if the calculation succeeded.

Related Properties and Methods

[App.CalculateSpectrum page 206](#)
[App.CalculateSpectrumFast page 206](#)
[App.CalculatingSpectrum page 192](#)
[App.DeleteSpectrumOnCalculate page 193](#)
[Calc.CalculateToEMin page 235](#)

Copy

Function Copy(A : IApp) : Boolean;

Description

A. OLE automation reference

Copies all data from *Simnra.App* object *A* to the current object. The data include experimental setup, calculational setup, target and foil structure, all spectra, etc. The following code copies all data from *App2* to *App1*:

```
var
  App1, App2 : Variant;
begin
  App1 := CreateOLEObject(Simnra.App');
  App2 := CreateOLEObject(Simnra.App');
  App1.Copy(App2);
end;
```

Parameters

A *Simnra.App* object.

Return Value

Returns *true* if ok.

Related Properties and Methods

Calc.Copy [page 245](#)
Foil.Copy [page 274](#)
Projectile.Copy [page 233](#)
Setup.Copy [page 228](#)
Spectrum.Copy [page 300](#)
Target.Copy [page 262](#)
Window.Copy [page 286](#)

CopySpectrumData

Procedure CopySpectrumData;

Description

Copies experimental and simulated spectra in ASCII format to the Windows clipboard. See [Edit: Copy Data](#) for a description of the format.

Parameters

None.

Return Value

None.

Related Properties and Methods

`App.WriteSpectrumData` [page 217](#)

CreateListOfCrSecs

Procedure `CreateListOfCrSecs`;

Description

Creates the list of available cross-section data files.

Parameters

None.

Return Value

None.

FitSpectrum

Function `FitSpectrum` : Boolean;

Description

Fits a spectrum. The fit parameters in the [Simnra.Fit](#) object have to be adjusted before the fit can be performed with *FitSpectrum*.

Parameters

None.

Return Value

Returns *true* if the fit succeeded.

Hide

Procedure `Hide`;

Description

A. OLE automation reference

Hides SIMNRA. The program is still running, but is not visible.

Parameters

None.

Return Value

None.

Related Properties and Methods

App.Minimize [page 210](#)

App.Show [page 216](#)

Maximize

Procedure Maximize;

Description

Maximizes SIMNRA to fill the whole screen. SIMNRA must be visible, i.e. not minimized or hidden, otherwise *Maximize* has no effect.

Parameters

None.

Return Value

None.

Related Properties and Methods

App.Active [page 191](#)

App.BringToFront [page 203](#)

App.Minimize [page 210](#)

App.Show [page 216](#)

Minimize

Procedure Minimize;

Description

Minimizes SIMNRA to the Windows task bar.

Parameters

None.

Return Value

None.

Related Properties and Methods

App.Active [page 191](#)

App.BringToFront [page 203](#)

App.Maximize [page 210](#)

App.Restore [page 215](#)

Open

Function Open(FileName : WideString(; FileFormat : Integer = -1)) : Boolean;

Description

Opens an XNRA, NRA or IDF-file. SIMNRA will try to determine the format of the file automatically from the file extension. A call to *Open* is identical to calling *OpenAs(FileName, -1)*.

Parameters

FileName The name of the file including path.

FileFormat This parameter is ignored. Nevertheless, a value has to be given for compatibility with SIMNRA 6.06.

Return Value

Returns *true* if the file was opened successfully.

Related Properties and Methods

App.OpenAs [page 212](#)

App.OpenStream [page 212](#)

App.SaveAs [page 215](#)

OpenAs

A. OLE automation reference

Function OpenAs(FileName : WideString (; FileFormat : Integer = -1)) : Boolean;

Description

Opens an XNRA, NRA or IDF-file. The parameter *FileFormat* is optional and can be omitted. If *FileFormat* is omitted, then SIMNRA will try to determine the format of the file automatically from the file extension. If *FileFormat* is given, then the file is assumed to be of that format.

Parameters

<i>FileName</i>	The name of the file including path.
<i>FileFormat</i>	Format of the file. Allowed values are: <ul style="list-style-type: none">-1 Unknown file format. This is the default if <i>FileFormat</i> is omitted. In this case SIMNRA will try to determine the format of the file automatically from the file extension.0 NRA file format, as used by SIMNRA 6 and earlier.1 IBA data format (IDF).2 XNRA file format, as used by SIMNRA 7 and later.

Return Value

Returns *true* if the file was opened successfully.

Related Properties and Methods

App.Open [page 211](#)

App.OpenStream [page 212](#)

App.SaveAs [page 215](#)

OpenStream

Function OpenStream(Stream: IUnknown; FileFormat: Integer) : Boolean;

Description

Opens an XNRA or NRA file from an *IStream* interface.

Parameters

<i>Stream</i>	<i>IStream</i> interface containing the file stream. The <i>IStream</i> interface has to be casted to a <i>IUnknown</i> interface.
<i>FileFormat</i>	Format of the file. Allowed values are: <ul style="list-style-type: none">0 NRA file format, as used by SIMNRA 6 and earlier.2 XNRA file format, as used by SIMNRA 7 and later.

Return Value

Returns *true* if the stream was read successfully.

Related Properties and Methods

App.Open [page 211](#)

App.OpenAs [page 212](#)

ReadSpectrumData

Function ReadSpectrumData(FileName : WideString; Format : Integer) : Boolean;

Description

Imports experimental data in different formats.

Parameters

<i>FileName</i>	The name of the spectrum data file including path.
<i>Format</i>	<p>Format of the spectrum data file, see subsection 3.3.5 for format details. Allowed values for <i>Format</i> are:</p> <ol style="list-style-type: none"> 1 Data in ASCII file format with channel numbers and counts. 2 Data in Canberra's CAM file format. 4 Data in RUMP's RBS file format. 5 Data in user defined format. Requires a user supplied dll. See section 3.16 for more details. 6 Data in MCERD format, calculated by the Monte-Carlo code MCERD. 7 Data in ASCII file format without channel numbers, identical to RUMP's 'swallow' command. 8 Data stored as energy – counts, with energy in keV. 9 Data from an XNRA or IDF file. 10 Data stored in Canberra's AVA file format. 11 Data stored in FAST ComTec's MPA file format. 12 Data stored in ASCII one-column format with counts. 13 Data stored in IAEA SPE spectrum format. 14 Spectrum data from a RUMP RBS file. <p>If the file contains multiple spectra then the first spectrum is imported.</p>

Return Value

Returns *true* if the file was imported successfully.

Related Properties and Methods

App.WriteSpectrumData [page 217](#)

Reset

Procedure Reset;

Description

Resets the program to its starting values. All calculated spectra, target, foil, window and setup definitions are deleted.

Parameters

None.

Return Value

None.

ResizeSpectrum

Procedure ResizeSpectrum(NumChannels : Integer; ResizeCalibration : Boolean);

Description

Resizes the experimental spectrum to *NumChannels* channels, see [subsection 3.9.6](#). Depending on *ResizeCalibration* the energy calibration can be adjusted to the resized spectrum.

Parameters

<i>NumChannels</i>	Desired number of channels of the spectrum.
<i>ResizeCalibration</i>	If <i>True</i> , then the detector energy calibration is adjusted to the resized spectrum and the simulated spectrum is recalculated. If <i>False</i> , then only the experimental spectrum is resized.

Return Value

None.

Restore

Procedure Restore;

Description

Restores the minimized application to its normal size.

Parameters

None.

Return Value

None.

Related Properties and Methods

App.Active [page 191](#)

App.BringToFront [page 203](#)

App.Minimize [page 210](#)

SaveAs

Function SaveAs(FileName : WideString; FileType : Integer = 2) : Boolean;

Description

Saves a NRA-file.

Parameters

FileName The name of the NRA-file including path. If the file already exists it will be overwritten.

FileType Format of the file. *FileType* is optional and can be omitted. It is used only for compatibility to SIMNRA 6.06, the value of *FileType* is not used. *SaveAs* always writes a file in XNRA file format.

Return Value

Returns *true* if the file was saved successfully.

Related Properties and Methods

App.Open [page 211](#)

SaveThumbnailAs

Procedure SaveThumbnailAs(FileName : WideString; Width : Integer);

Description

Saves a thumbnail of the current chart as bitmap file.

Parameters

<i>FileName</i>	The name of the bitmap file including path. The file extension should be <i>.bmp</i> . If the file already exists it will be overwritten.
<i>Width</i>	Width of the thumbnail image, in pixels. The height of the image will be adjusted automatically.

Return Value

None.

Show

Procedure Show;

Description

Shows SIMNRA, if it was hidden.

Parameters

None.

Return Value

None.

Related Properties and Methods

App.Hide [page 209](#)

WriteSpectrumData

Function WriteSpectrumData(FileName : WideString) : Boolean;

Description

Writes all spectra (experimental, simulated) in ASCII-format to a file. See [subsection 3.3.6](#) for a description of the file format.

Parameters

<i>FileName</i>	Name of the data file including path. If the file already exists it will be overwritten.
-----------------	--

Return Value

Returns *true* if the file was written successfully.

Related Properties and Methods

App.ReadSpectrumData [page 213](#)

App.CopySpectrumData [page 208](#)

A.6. *Simnra.Setup*

The *Simnra.Setup* object represents the experimental setup, excluding the incident projectile. The projectile is represented by [Simnra.Projectile](#).

A.6.1. Properties

Alpha

[Get/Set] Property Alpha : Double;

Description

Incident angle α [deg].

Related Properties and Methods

Setup.Beta [page 221](#)

Setup.Theta [page 227](#)

ApertureDistance

[Get/Set] Property ApertureDistance : Double;

Description

Distance between sample and detector aperture [mm], see [subsubsection 3.5.2](#).

Related Properties and Methods

Setup.ApertureHeight [page 218](#)

Setup.ApertureShape [page 218](#)

Setup.ApertureWidth [page 219](#)

Setup.BeamHeight [page 219](#)

Setup.BeamShape [page 220](#)

Setup.BeamWidth [page 220](#)

ApertureHeight

[Get/Set] Property ApertureHeight : Double;

Description

Height of the detector aperture [mm], see [subsubsection 3.5.2](#).

Related Properties and Methods

Setup.ApertureDistance [page 217](#)

Setup.ApertureShape [page 218](#)

Setup.ApertureWidth [page 219](#)

Setup.BeamHeight [page 219](#)

Setup.BeamShape [page 220](#)

Setup.BeamWidth [page 220](#)

ApertureShape

[Get/Set] Property ApertureShape : Integer;

Description

Shape of the detector aperture, see [subsubsection 3.5.2](#). Possible values are:

0 : Circular

1 : Rectangular.

Related Properties and Methods

Setup.ApertureDistance [page 217](#)
Setup.ApertureHeight [page 218](#)
Setup.ApertureWidth [page 219](#)
Setup.BeamHeight [page 219](#)
Setup.BeamShape [page 220](#)
Setup.BeamWidth [page 220](#)

ApertureWidth

[Get/Set] Property ApertureWidth : Double;

Description

Width of the detector aperture [mm], see [subsection 3.5.2](#).

Related Properties and Methods

Setup.ApertureDistance [page 217](#)
Setup.ApertureHeight [page 218](#)
Setup.ApertureShape [page 218](#)
Setup.BeamHeight [page 219](#)
Setup.BeamShape [page 220](#)
Setup.BeamWidth [page 220](#)

BeamHeight

[Get/Set] Property BeamHeight : Double;

Description

Height of the incident beam [mm], see [subsection 3.5.2](#).

Related Properties and Methods

Setup.ApertureDistance [page 217](#)
Setup.ApertureHeight [page 218](#)
Setup.ApertureShape [page 218](#)
Setup.ApertureWidth [page 219](#)
Setup.BeamShape [page 220](#)
Setup.BeamWidth [page 220](#)

BeamShape

A. OLE automation reference

[Get/Set] Property BeamShape : Integer;

Description

Shape of the incident beam, see [subsubsection 3.5.2](#). Possible values are:

0 : Circular

1 : Rectangular.

Related Properties and Methods

Setup.ApertureDistance [page 217](#)

Setup.ApertureHeight [page 218](#)

Setup.ApertureShape [page 218](#)

Setup.ApertureWidth [page 219](#)

Setup.BeamHeight [page 219](#)

Setup.BeamWidth [page 220](#)

Beamspread

[Get/Set] Property Beamspread : Double;

Description

Energy spread of incident beam [keV FWHM].

Related Properties and Methods

Setup.Energy [page 224](#)

BeamWidth

[Get/Set] Property BeamWidth : Double;

Description

Width of the incident beam [mm], see [subsubsection 3.5.2](#).

Related Properties and Methods

Setup.ApertureDistance [page 217](#)
Setup.ApertureHeight [page 218](#)
Setup.ApertureShape [page 218](#)
Setup.ApertureWidth [page 219](#)
Setup.BeamHeight [page 219](#)
Setup.BeamShape [page 220](#)

Beta

[Get/Set] Property Beta : Double;

Description

Exit angle β [deg].

Related Properties and Methods

Setup.Alpha [page 217](#)
Setup.SetBeta [page 231](#)
Setup.Theta [page 227](#)

CalibrationLinear

[Get/Set] Property CalibrationLinear : Double;

Description

Linear calibration term B for energy calibration, see [Equation 3.1](#) [keV/channel].

Related Properties and Methods

Setup.CalibrationOffset [page 221](#)
Setup.CalibrationQuadratic [page 222](#)

CalibrationOffset

[Get/Set] Property CalibrationOffset : Double;

Description

Calibration offset A for energy calibration, see [Equation 3.1](#) [keV].

Related Properties and Methods

Setup.CalibrationLinear [page 221](#)

Setup.CalibrationQuadratic [page 222](#)

CalibrationQuadratic

[Get/Set] Property CalibrationQuadratic : Double;

Description

Quadratic calibration term C for energy calibration, see [Equation 3.1](#) [keV/channel²].

Related Properties and Methods

Setup.CalibrationLinear [page 221](#)

Setup.CalibrationOffset [page 221](#)

DetectorResolution

[Get/Set] Property DetectorResolution : Double;

Description

Detector resolution [keV FWHM].

DetectorPosition

[Get/Set] Property DetectorPosition : Integer;

Description

Position of the detector.

Possible values are:

0 : IBM geometry.

1 : Cornell geometry.

2 : General geometry.

SIMNRA will try to detect automatically the correct geometry from incident angle, scattering angle and exit angle. However, in some cases (for example normal incidence) this is ambiguous. In these cases the position of the detector should be given. This plays only a role for geometrical straggling and multiple scattering calculations.

DetectorSensitivity

[Get/Set] Property DetectorSensitivity : Integer;

Description

Returns if the detector sensitivity is equal to unity or has been read from file; allows to set the detector sensitivity to unity.

Possible return values are:

1 : The detector sensitivity is equal to unity.

2 : The detector sensitivity has been read from file. The name of the file can be retrieved by [Setup.DetectorSensitivityFileName](#).

Possible set values are:

1 : The detector sensitivity is set to unity.

Other values are not allowed. To read a file with detector sensitivity data [Setup.ReadDetectorSensitivity](#) has to be used.

Related Properties and Methods

Setup.DetectorSensitivityFileName [page 223](#)

Setup.ReadDetectorSensitivity [page 230](#)

DetectorSensitivityFileName

[Get] Property DetectorSensitivityFileName : WideString;

Description

Returns the name of the detector sensitivity file if the detector sensitivity has been read from file. Returns an empty string if the detector sensitivity is unity.

DetectorSensitivityFileName cannot be used to read a file, use [Setup.ReadDetectorSensitivity](#) for reading a file.

Related Properties and Methods

Setup.DetectorSensitivity [page 223](#)

Setup.ReadDetectorSensitivity [page 230](#)

DetectorType

[Get/Set] Property DetectorType : Integer;

Description

A. OLE automation reference

Type of detector. Allowed values are:

0 : Solid-state detector.

1 : Time-of-flight detector.

2 : Electrostatic detector.

Related Properties and Methods

Setup.DetectorResolution [page 222](#)

Energy

[Get/Set] Property Energy : Double;

Description

Energy of incident ions [keV].

Related Properties and Methods

Setup.Beamsread [page 220](#)

LiveTime

[Get/Set] Property LiveTime : Double;

Description

Live time of a measurement [s].

Related Properties and Methods

Setup.LTCorrection [page 224](#)

Setup.RealTime [page 226](#)

LTCorrection

[Get/Set] Property LTCorrection : Boolean;

Description

True if a live time correction is applied, else *false*.

Related Properties and Methods

Setup.LiveTime [page 224](#)

Setup.RealTime [page 226](#)

ParticlesSr

[Get/Set] Property ParticlesSr : Double;

Description

Number of incident particles times solid angle [sr].

PUCalculation

[Get/Set] Property PUCalculation : Boolean;

Description

True if pile-up is calculated, else *false*.

Related Properties and Methods

Setup.PUROn [page 225](#)

Setup.PURResolution [page 226](#)

Setup.RiseTime [page 226](#)

PUROn

[Get/Set] Property PUROn : Boolean;

Description

True if a pile-up rejector was used, else *false*. See [Pile-up rejector](#) in [subsubsection 3.5.2](#) for details.

Related Properties and Methods

Setup.PUCalculation [page 225](#)

Setup.PURResolution [page 226](#)

Setup.RiseTime [page 226](#)

PURResolution

[Get/Set] Property PURResolution : Double;

Description

Pile-up rejector pair-resolution time [μ s]. See [Pile-up rejector pair-resolution time](#) in [subsubsection 3.5.2](#) for details.

Related Properties and Methods

Setup.PUCalculation [page 225](#)

Setup.PUROn [page 225](#)

Setup.RiseTime [page 226](#)

RealTime

[Get/Set] Property RealTime : Double;

Description

Real time of a measurement [s].

Related Properties and Methods

Setup.LiveTime [page 224](#)

Setup.LTCorrection [page 224](#)

RiseTime

[Get/Set] Property RiseTime : Double;

Description

Rise time of the amplified pulse from zero to its maximum value [μ s]. See [subsubsection 3.5.2](#) for details.

Related Properties and Methods

Setup.PUCalculation [page 225](#)

Setup.PURResolution [page 226](#)

Theta

[Get/Set] Property Theta : Double;

Description

Scattering angle θ [deg].

Related Properties and Methods

Setup.Alpha [page 217](#)

Setup.Beta [page 221](#)

TOFLength

[Get/Set] Property TOFLength : Double;

Description

Length of the flight path for a time-of-flight detector [m]. This value is only used, if [Setup.DetectorType](#) is set to time-of-flight detector.

Related Properties and Methods

Setup.DetectorType [page 223](#)

Setup.TOFTimeResolution [page 227](#)

TOFTimeResolution

[Get/Set] Property TOFTimeResolution : Double;

Description

Time resolution of a time-of-flight detector [ps]. The full width at half maximum (FWHM) has to be used. This value is only used, if [Setup.DetectorType](#) is set to time-of-flight detector.

Related Properties and Methods

Setup.DetectorType [page 223](#)

Setup.TOFLength [page 227](#)

A.6.2. Methods**Copy**

Function Copy(S : ISetup) : Boolean;

A. OLE automation reference

Description

Copies all data (including projectile) from SIMNRA experimental setup *S* to the current setup.

The following code copies all data from *Setup2* to *Setup1*:

```
var
  Setup1, Setup2 : Variant;
begin
  Setup1 := CreateOLEObject(Simnra.Setup');
  Setup2 := CreateOLEObject(Simnra.Setup');
  Setup1.Copy(Setup2);
end;
```

Parameters

S *Simnra.Setup* experimental setup object.

Return Value

Returns *true* if ok.

Related Properties and Methods

App.Copy [page 207](#)
Calc.Copy [page 245](#)
Foil.Copy [page 274](#)
Projectile.Copy [page 233](#)
Spectrum.Copy [page 300](#)
Target.Copy [page 262](#)
Window.Copy [page 286](#)

Open

Function Open(FileName : WideString) : Boolean;

Description

Reads the experimental setup from an XNRA, NRA or IDF-file. SIMNRA will try to determine the format of the file automatically from the file extension. A call to *Open* is identical to calling *Setup.OpenAs(FileName, -1)*.

Parameters

FileName The name of the file including path.

Return Value

Returns *true* if the file was read successfully.

Related Properties and Methods

Setup.OpenAs [page 229](#)

Setup.SaveAs [page 230](#)

App.Open [page 211](#)

OpenAs

Function OpenAs(FileName : WideString (; FileFormat : Integer = -1)) : Boolean;

Description

Reads the experimental setup from an XNRA, NRA or IDF-file. The parameter *FileFormat* is optional and can be omitted. If *FileFormat* is omitted, then SIMNRA will try to determine the format of the file automatically from the file extension. If *FileFormat* is given, then the file is assumed to be of that format.

Parameters

<i>FileName</i>	The name of the file including path.
<i>FileFormat</i>	Format of the file. Allowed values are: <ul style="list-style-type: none"> -1 Unknown file format. This is the default if <i>FileFormat</i> is omitted. In this case SIMNRA will try to determine the format of the file automatically from the file extension. 0 NRA file format, as used by SIMNRA 6 and earlier. 1 IBA data format (IDF). 2 XNRA file format, as used by SIMNRA 7 and later.

Return Value

Returns *true* if the file was read successfully.

Related Properties and Methods

Setup.Open [page 228](#)

Setup.SaveAs [page 230](#)

App.OpenAs [page 212](#)

ReadDetectorSensitivity

Function ReadDetectorSensitivity(FileName : WideString; Format : Integer) : Boolean;

A. OLE automation reference

Description

Reads the detector sensitivity as function of energy from file, see [section 3.5.2](#) for details.

Parameters

<i>FileName</i>	Name of the file including path.
<i>Format</i>	Format of the detector sensitivity file. Possible values are: 1 : xnra file. 2 : xml or idf file. 3 : ASCII file.

Return Value

Returns *true* if the file was read successfully.

Related Properties and Methods

Setup.DetectorSensitivity [page 223](#)

Setup.DetectorSensitivityFileName [page 223](#)

SaveAs

Function SaveAs(FileName : WideString) : Boolean;

Description

Saves the current experimental setup (including projectile) to the file *FileName*.

Parameters

<i>FileName</i>	Name of the file including path. The file extension should be either <i>.xsetup</i> or <i>.xnra</i> .
-----------------	---

Return Value

Returns *true* if the file was saved successfully.

Related Properties and Methods

Setup.SaveAsDefault [page 231](#)

App.SaveAs [page 215](#)

SaveAsDefault

Function SaveAsDefault : Boolean;

Description

Saves the current experimental setup as default for all future simulations.

Parameters

None.

Return Value

Returns *true* if the current experimental setup was saved successfully as default.

Related Properties and Methods

Setup.SaveAs [page 230](#)

Calc.SaveAsDefault [page 247](#)

SetBeta

Function SetBeta(Geometry : Integer) : Double;

Description

Sets the exit angle β as function of incident angle α , scattering angle θ and scattering geometry. The function returns the new value of β [deg]. An angle of -90° indicates an error.

Parameters

<i>Geometry</i>	Scattering geometry. Possible values are: 0 : IBM geometry. 1 : Cornell geometry.
-----------------	---

Return Value

Returns the new value of β [deg].

Related Properties and Methods

Setup.Alpha [page 217](#)

Setup.Beta [page 221](#)

Setup.Theta [page 227](#)

A.7. Simnra.Projectile

The *Simnra.Projectile* object represents the incident projectile.

A.7.1. Properties

Charge

[Get/Set] Property Charge : Integer;

Description

Nuclear charge (or atomic number) of the projectile. If the nuclear charge is set, then the mass of the projectile is set automatically to the mass of the isotope with the highest natural abundance. Use [Projectile.Mass](#) to change the mass of the projectile.

Related Properties and Methods

Projectile.Name page 233

Projectile.Mass page 232

Mass

[Get/Set] Property Mass : Double;

Description

Mass of the projectile [amu]. If the mass is set, then SIMNRA will replace the given mass by the precise mass of the isotope. If you want to change the mass, then you have to set the nuclear charge of the projectile or the projectile name first.

Example:

```
Projectile.Name := 'He'; { Projectile  $^4\text{He}$  is selected, because the isotope  $^4\text{He}$   
has the highest natural abundance }
```

```
Projectile.Mass := 3;      { Mass 3.016029 amu is selected and set }
```

Related Properties and Methods

Projectile.Charge page 232

Projectile.Name page 233

Name

[Get/Set] Property Name : WideString;

Description

Name (element name) of the projectile. The name should consist of two letters, such as 'He' or 'H'. If the name is set, then the mass of the projectile is set automatically to the mass of the isotope with the highest natural abundance. Use [Projectile.Mass](#) to change the mass of the projectile.

Related Properties and Methods

Projectile.Charge [page 232](#)

Projectile.Mass [page 232](#)

A.7.2. Methods

Copy

Function Copy(*P* : IProjectile) : Boolean;

Description

Copies the projectile data from SIMNRA projectile *P* to the current projectile.

Note that [Setup.Copy](#) copies all experimental setup data including all projectile data from one experimental setup to another, i.e. when using [Setup.Copy](#) then *Projectile.Copy* is not further necessary.

The following code copies all data from *Projectile2* to *Projectile1*:

```
var
  Projectile1, Projectile2 : Variant;
begin
  Projectile1 := CreateOLEObject('Simnra.Projectile');
  Projectile2 := CreateOLEObject('Simnra.Projectile');
  Projectile1.Copy(Projectile2);
end;
```

Parameters

P *Simnra.Projectile* projectile object.

Return Value

Returns *true* if ok.

Related Properties and Methods

A. OLE automation reference

App.Copy [page 207](#)
Calc.Copy [page 245](#)
Foil.Copy [page 274](#)
Setup.Copy [page 228](#)
Spectrum.Copy [page 300](#)
Target.Copy [page 262](#)
Window.Copy [page 286](#)

A.8. Simnra.Calc

The *Simnra.Calc* object represents the parameters of the calculation.

A.8.1. Properties

Accuracy

[Get] Property Accuracy : Integer;

Description

Specifies the accuracy of a calculation. *Accaracy* is either *Accurate* if the calculation was performed by clicking [Calculate spectrum](#) or by using the OLE method [App.CalculateSpectrum](#), or *Fast* if the calculation was performed by clicking [Calculate spectrum fast](#) or using the OLE method [App.CalculateSpectrumFast](#). *Accuracy* is readonly and can be used only to get the accuracy of a previous calculation. The accuracy of a calculation is set when calculating a spectrum as described above, see [section 3.8](#) for details.

Possible values for *Accuracy* are:

0 : *Accurate*

1 : *Fast*

Related Properties and Methods

App.CalculateSpectrum [page 206](#)

App.CalculateSpectrumFast [page 206](#)

AutoStepwidthIn

[Get/Set] Property AutoStepwidthIn : Boolean;

Description

Specifies if automatic step width control for incident ions is used.

Related Properties and MethodsCalc.AutoStepwidthOut [page 235](#)

AutoStepwidthOut

[Get/Set] Property AutoStepwidthOut : Boolean;

Description

Specifies if automatic step width control for outgoing ions is used.

Related Properties and MethodsCalc.AutoStepwidthIn [page 234](#)

CalculateToEMin

[Get/Set] Property CalculateToEMin : Boolean;

Default Value

false

Description

If *CalculateToEMin* is *false*, then SIMNRA speeds up calculations by terminating a calculation if backscattered or recoiled particles do not reach the detector any more, i.e. if the energy of backscattered or recoiled particles at the detector has decreased below the cutoff energy *Calc.EMin*. If *CalculateToEMin* is set to *true*, then calculations are proceeded until the energy of incident particles has decreased below *Calc.EMin*. Setting *CalculateToEMin* to *true* may slow down calculations considerably.

If the function *App.CalculateSpectrumToDepth(Depth)* is used together with *CalculateToEMin = true*, then the *Depth* value has priority and the calculation is terminated when the depth of incident ions has reached *Depth* independent of their energy.

Related Properties and MethodsCalc.EMin [page 238](#)App.CalculateSpectrumToDepth [page 207](#)

CreateSpectrum

A. OLE automation reference

[Get/Set] Property CreateSpectrum : Boolean;

Default Value

true

Description

Specifies if a spectrum is calculated or not. This parameter is always *true*, and should not be changed.

CreateSpectrumFromLayerNr

[Get/Set] Property CreateSpectrumFromLayerNr : Integer;

Default Value

0

Description

Specifies that a spectrum is calculated only from one layer in the target. The number of this layer is given in *CreateSpectrumFromLayerNr*. Other layers are still present and act for example through their stopping powers, but there are no particles backscattered from these layers. If *CreateSpectrumFromLayerNr* is ≤ 0 then the spectrum from all layers is calculated.

CrossSecStraggling

[Get/Set] Property CrossSecStraggling : Integer;

Description

Specifies how the cross-section and straggling are treated. Allowed values are:
1 : *Fast* model. Straggling is not taken into account for calculating the cross-section.
2 : *Accurate* model. The mean cross-section weighted with the straggling distribution is used.

dEin

[Get/Set] Property dEin : Double;

Description

Stepwidth incident ions [keV].

Related Properties and Methods

Calc.dEout [page 237](#)

dEout

[Get/Set] Property dEout : Double;

Description

Stepwidth outgoing ions [keV].

Related Properties and Methods

Calc.dEin [page 236](#)

DualScattering

[Get/Set] Property DualScattering : Boolean;

Description

Specifies if dual scattering is calculated.

DualScatteringRoughness

[Get/Set] Property DualScatteringRoughness : Integer;

Description

Selects how dual scattering and roughness are calculated simultaneously. Allowed values are:

0 : Accurate calculation of simultaneous dual scattering and roughness. This increases the computing time considerably.

1 : Fast calculation of simultaneous dual scattering and roughness.

This switch has only influence, if dual scattering is selected and the target contains rough layers or a rough substrate. See [Calculation of dual scattering and roughness](#) in [subsubsection 3.5.3](#) for more details.

Related Properties and Methods

Calc.DualScattering [page 237](#)

ElementSpectra

[Get/Set] Property ElementSpectra : Boolean;

Default Value

true

Description

Specifies if individual spectra for each element in the target are calculated.

Related Properties and Methods

Calc.IsotopeSpectra [page 239](#)

EMin

[Get/Set] Property EMin : Double;

Description

Cutoff energy [keV].

GeometricalStraggling

[Get/Set] Property GeometricalStraggling : Boolean;

Default Value

false

Description

Specifies if energy straggling by geometrical spread is calculated.

Note: If *Calc.Straggling* is *false* then *GeometricalStraggling* has no effect.

Related Properties and Methods

[Calc.MultipleScattering page 240](#)
[Calc.Straggling page 243](#)
[Setup.ApertureDistance page 217](#)
[Setup.ApertureHeight page 218](#)
[Setup.ApertureShape page 218](#)
[Setup.ApertureWidth page 219](#)
[Setup.BeamHeight page 219](#)
[Setup.BeamShape page 220](#)
[Setup.BeamWidth page 220](#)

HighEnergyStopping

[Get/Set] Property HighEnergyStopping : Boolean;

Description

Selects if high energy stopping power data are used or not. This switch is only used together with the stopping power data by Andersen-Ziegler and has no influence if any other stopping power data set is selected. See [subsection 3.5.3](#) for more details.

Related Properties and Methods

[Calc.StoppingModel page 243](#)
[Calc.ZBStopping page 244](#)

Isotopes

[Get/Set] Property Isotopes : Boolean;

Description

Specifies if isotopes are taken into account.

IsotopeSpectra

[Get/Set] Property IsotopeSpectra : Boolean;

Default Value

true

Description

A. OLE automation reference

Specifies if individual spectra for each isotope in the target are calculated.

Related Properties and Methods

Calc.ElementSpectra [page 238](#)

LogFile

[Get/Set] Property LogFile : Boolean;

Description

Specifies if a log file with additional information about the calculation is created. The name of the log file is *SIMNRA.LOG*. It is stored in the directory where the last SIMNRA data file was opened or saved.

MultipleScattering

[Get/Set] Property MultipleScattering : Boolean;

Default Value

false

Description

Specifies if energy spread by multiple scattering is calculated.

Note: If *Calc.Straggling* is false then *MultipleScattering* has no effect.

Related Properties and Methods

Calc.GeometricalStraggling [page 238](#)

Calc.MultipleScatteringModel [page 240](#)

Calc.Straggling [page 243](#)

MultipleScatteringModel

[Get/Set] Property MultipleScatteringModel : Integer;

Default Value

1

Description

Selects the multiple scattering model. Possible values are:

0 : Szilagy model.

1 : Amsel model.

Note: For activating multiple scattering in calculations *Calc.MultipleScattering* must be set to *true*. *Calc.MultipleScatteringModel* selects the applied model, but does not activate multiple scattering.

Related Properties and Methods

Calc.MultipleScattering [page 240](#)

Calc.Straggling [page 243](#)

NuclearStoppingModel

[Get/Set] Property NuclearStoppingModel : Integer;

Description

Selects the nuclear stopping power model. Allowed values are:

0 : No nuclear stopping.

1 : Universal nuclear stopping.

See [subsubsection 3.5.3](#) and [subsection 4.6.5](#) for more details about nuclear stopping power models.

NumberOfAngleVariations

[Get/Set] Property NumberOfAngleVariations : Integer;

Description

Number of angle steps in the calculation of rough substrates.

Related Properties and Methods

Calc.NumberOfDVariations [page 242](#)

NumberOfDVariations

[Get/Set] Property NumberOfDVariations : Integer;

Description

Number of thickness steps in the calculation of rough layers.

Related Properties and Methods

Calc.NumberOfAngleVariations [page 241](#)

PUModel

[Get/Set] Property PUModel : Integer;

Description

Selects the [pile-up model](#), see [subsubsection 3.5.3](#). Allowed values are:

0 : *Accurate* model

1 : *Fast* model.

ScreeningModel

[Get/Set] Property ScreeningModel : Integer;

Description

Selects the screening function to the Rutherford cross-section due to partial screening of the nuclear charges by the electron shells surrounding both nuclei, see [section 4.3](#). Allowed values are:

0 : Rutherford cross-section without screening ([Equation 4.11](#))

1 : Andersen's screening function ([Equation 4.14](#)).

2 : L'Ecuyer screening function ([Equation 4.13](#)).

3 : Universal screening function.

Rutherford cross-section without screening should be used only for test purposes. Andersen's screening function is very accurate at sufficiently high energies and the program default. The Universal screening function is more accurate than Andersen screening at low energies, see [section 4.3](#).

Related Properties and Methods

Simnra.CrossSec object [page 314](#)

CrossSec.RBSRutherford [page 319](#)

StoppingModel

[Get/Set] Property StoppingModel : Integer;

Description

Selects the electronic stopping power model. Allowed values are:

0 : Andersen-Ziegler stopping.

1 : Ziegler-Biersack stopping.

2 : KKKNS stopping.

3 : User defined stopping.

4 : SRIM stopping.

See [subsubsection 3.5.3](#) and [section 4.6](#) for more details about stopping power models.

Straggling

[Get/Set] Property Straggling : Boolean;

Description

Specifies if energy straggling in window, target and foil is taken into account. If *Straggling* is set to *true*, then energy-loss straggling is taken into account. Geometrical straggling can be taken additionally into account if [Calc.GeometricalStraggling](#) is set to *true*, energy straggling by multiple scattering is taken into account if [Calc.MultipleScattering](#) is set to *true*. If *Straggling* is set to *false*, then all energy straggling components (energy-loss straggling, geometrical straggling, energy straggling by multiple scattering) are neglected independent of the settings of [Calc.GeometricalStraggling](#) and [Calc.MultipleScattering](#).

Related Properties and Methods

[Calc.GeometricalStraggling](#) [page 238](#)

[Calc.MultipleScattering](#) [page 240](#)

[Calc.StragglingModel](#) [page 243](#)

StragglingModel

[Get/Set] Property StragglingModel : Integer;

Description

A. OLE automation reference

Selects the electronic energy-loss straggling model. Allowed values are:

0 : No straggling.

1 : Bohr's theory.

2 : Chu's theory.

3 : Yang's theory.

See [subsection 4.8.2](#) for more details about straggling models.

Related Properties and Methods

Calc.Straggling [page 243](#)

StragglingShape

[Get/Set] Property StragglingShape : Integer;

Description

Selects the shape of straggling distributions. Allowed values are:

0 : Gaussian shape.

1 : Asymmetric Gaussian shape.

See [subsection 4.8.2](#) for more details about straggling models.

Related Properties and Methods

Calc.Straggling [page 243](#)

SubstrateRoughnessDimension

[Get/Set] Property SubstrateRoughnessDimension : Integer;

Description

Dimensionality of substrate roughness. Allowed values are:

0: Dimension 2.0

1: Dimension 2.5

ZBStopping

[Get/Set] Property ZBStopping : Boolean;

Description

Selects Ziegler-Biersack or Andersen-Ziegler stopping. If *ZBStopping* is *true*, Ziegler-Biersack stopping is used, while Andersen-Ziegler is selected if *ZBStopping* is set to *false*. See [subsection 3.5.3](#) for more details about stopping power data.

Note: *ZBStopping* is deprecated and only supported for backward compatibility. It should not be used for new projects. Use [Calc.StoppingModel](#) instead.

Related Properties and Methods

Calc.HighEnergyStopping [page 239](#)

Calc.StoppingModel [page 243](#)

A.8.2. Methods

Copy

Function Copy(*C* : ICalc) : Boolean;

Description

Copies all data from SIMNRA calculational setup *C* to the current calculational setup.

The following code copies all data from *Calc2* to *Calc1*:

```
var
  Calc1, Calc2 : Variant;
begin
  Calc1 := CreateOLEObject(Simnra.Calc');
  Calc2 := CreateOLEObject(Simnra.Calc');
  Calc1.Copy(Calc2);
end;
```

Parameters

C *Simnra.Calc* calculational setup object.

Return Value

Returns *true* if ok.

Related Properties and Methods

App.Copy [page 207](#)

Projectile.Copy [page 233](#)

Setup.Copy [page 228](#)

Open

Function Open(FileName : WideString) : Boolean;

Description

Reads the calculational setup from an XNRA, NRA or IDF-file. SIMNRA will try to determine the format of the file automatically from the file extension. A call to *Open* is identical to calling *Calc.OpenAs(FileName, -1)*.

Parameters

FileName The name of the file including path.

Return Value

Returns *true* if the file was read successfully.

Related Properties and Methods

App.Copy [page 207](#)
Foil.Copy [page 274](#)
Projectile.Copy [page 233](#)
Setup.Copy [page 228](#)
Spectrum.Copy [page 300](#)
Target.Copy [page 262](#)
Window.Copy [page 286](#)

OpenAs

Function OpenAs(FileName : WideString (; FileFormat : Integer = -1)) : Boolean;

Description

Reads the calculational setup from an XNRA, NRA or IDF-file. The parameter *FileFormat* is optional and can be omitted. If *FileFormat* is omitted, then SIMNRA will try to determine the format of the file automatically from the file extension. If *FileFormat* is given, then the file is assumed to be of that format.

Parameters

FileName The name of the file including path.

<i>FileFormat</i>	Format of the file. Allowed values are:
-1	Unknown file format. This is the default if <i>FileFormat</i> is omitted. In this case SIMNRA will try to determine the format of the file automatically from the file extension.
0	NRA file format, as used by SIMNRA 6 and earlier.
1	IBA data format (IDF).
2	XNRA file format, as used by SIMNRA 7 and later.

Return Value

Returns *true* if the file was read successfully.

Related Properties and Methods

Calc.Open [page 246](#)

Calc.SaveAs [page 247](#)

App.OpenAs [page 212](#)

SaveAs

Function SaveAs(FileName : WideString) : Boolean;

Description

Saves the current calculational setup to the file *FileName*.

Parameters

<i>FileName</i>	Name of the file including path. The file extension should be either <i>.xcalc</i> or <i>.xnra</i> .
-----------------	--

Return Value

Returns *true* if the file was saved successfully.

Related Properties and Methods

Calc.SaveAsDefault [page 247](#)

App.SaveAs [page 215](#)

SaveAsDefault

Function SaveAsDefault : Boolean;

Description

A. OLE automation reference

Saves the current calculational setup as default for all future simulations.

Parameters

None.

Return Value

Returns *true* if the current calculational setup was saved successfully as default.

Related Properties and Methods

Calc.SaveAs [page 247](#)

Setup.SaveAsDefault [page 231](#)

A.9. Simnra.Target

The *Simnra.Target* object represents the target with all layers and elements.

A.9.1. Properties

ElementAmount

[Get/Set] Property ElementAmount[lay, el : Integer] : Double;

Description

Amount of element number *el* in layer number *lay* [10^{15} atoms/cm²].

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq Target.NumberOfLayers$.
<i>el</i>	Number of the element, with $1 \leq el \leq Target.NumberOfElements[lay]$.

Related Properties and Methods

Target.ElementConcentration [page 249](#)

Target.LayerThickness [page 254](#)

ElementConcentration

[Get/Set] Property ElementConcentration[lay, el : Integer] : Double;

Description

Concentration of element number *el* in layer number *lay*. The sum of concentrations of all elements in a layer must be equal to 1. This is not checked by SIMNRA. If *ElementConcentration* is changed, it is the responsibility of the programmer to assure that the sum of the concentrations is equal to 1.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq \text{Target.NumberOfLayers}$.
<i>el</i>	Number of the element, with $1 \leq el \leq \text{Target.get.NumberOfElements}[lay]$.

Related Properties and Methods

Target.AddElement [page 259](#)
 Target.ElementName [page 250](#)
 Target.ElementZ [page 250](#)

ElementConcentrationArray

[Get/Set] Property ElementConcentrationArray[lay : Integer] : PSafeArray;

Description

Concentrations of all elements in layer number *lay* as a SafeArray data structure. Each element in the SafeArray is of type Double. The SafeArray is wrapped into a Variant array for compatibility with OLE automation.

The sum of concentrations of all elements in a layer must be equal to 1. This is not checked by SIMNRA. If *ElementConcentrationArray* is used to change element concentration, then it is the responsibility of the programmer to assure that the sum of the concentrations is equal to 1.

When using *ElementConcentrationArray* for setting element properties, then the number of elements in *ElementConcentrationArray* must be identical to the number of already existing elements in layer *lay*, i.e. *ElementConcentrationArray* does not add elements to a layer: This must be done beforehand using [Target.AddElement](#) or [Target.AddElements](#).

See [subsection A.18.3](#) for an example how to use *ElementConcentrationArray*.

ElementConcentrationArray is faster than successive calls to [Target.get.ElementConcentration](#) when used for multiple elements.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq \text{Target.NumberOfLayers}$.
------------	--

Related Properties and Methods

Target.AddElements [page 260](#)

Target.ElementConcentration [page 249](#)

Target.ElementZArray [page 251](#)

ElementName

[Get/Set] Property ElementName[lay, el : Integer] : WideString;

Description

Name of element number *el* in layer number *lay*. Element names are returned in standardized two-letters form with the first letter capitalized and the second letter in lowercase, e.g. 'Si'. One-letter names are returned with a blank as second letter, e.g. 'H '. 'XX' is returned if the element is unknown. For setting element names uppercase and lowercase is ignored and blanks are removed, i.e. silicon can be set with 'Si', 'SI', 'si', ' Si' or 'S i'.

Elements initially have isotopes with natural isotopic concentrations. Isotopic concentrations can be changed using the [IsotopeConcentration](#) property.

If *ElementName* is used to add new elements to the target (i.e. elements not already present in the target), then they will have undefined cross sections. The methods of the [Simmra.CrossSec](#) object have to be used to define the cross-section data.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq \text{Target.NumberOfLayers}$.
<i>el</i>	Number of the element, with $1 \leq el \leq \text{Target.NumberOfElements}[lay]$.

Related Properties and Methods

Target.AddElement [page 259](#)

Target.ElementConcentration [page 249](#)

Target.ElementZ [page 250](#)

ElementZ

[Get/Set] Property ElementZ[lay, el : Integer] : Integer;

Description

Nuclear Charge Z of element number el in layer number lay . Returns -1 if the element is unknown.

If *ElementZ* is used to add new elements to the target (i.e. elements not already present in the target), then they will have undefined cross sections. The methods of the *Simnra.CrossSec* object have to be used to define the cross-section data.

Parameters

lay Number of the layer, with $1 \leq lay \leq Target.NumberOfLayers$.

el Number of the element, with $1 \leq el \leq Target.NumberOfElements[lay]$.

Related Properties and Methods

Target.AddElement [page 259](#)

Target.ElementConcentration [page 249](#)

Target.ElementName [page 250](#)

ElementZArray

[Get/Set] Property ElementZArray[lay : Integer] : PSafeArray;

Description

Nuclear Charges Z of all elements in layer number lay as a SafeArray data structure. Each element in the SafeArray is of type Integer. The SafeArray is wrapped into a Variant array for compatibility with OLE automation. Returns -1 if the element is unknown.

When using *ElementZArray* for setting element properties, then the number of elements in *ElementZArray* must be identical to the number of already existing elements in layer lay , i.e. *ElementZArray* does not add elements to a layer: This must be done beforehand using *Target.AddElement* or *Target.AddElements*.

See [subsection A.18.3](#) for an example how to use *ElementZArray*.

ElementZArray is faster than successive calls to *Target.ElementZ* when used for multiple elements.

If *ElementZArray* is used to add new elements to the target (i.e. elements not already present in the target), then they will have undefined cross sections. The methods of the *Simnra.CrossSec* object have to be used to define the cross-section data.

Parameters

lay Number of the layer, with $1 \leq lay \leq Target.NumberOfLayers$.

Related Properties and Methods

A. OLE automation reference

Target.AddElements [page 260](#)
Target.ElementConcentrationArray [page 249](#)
Target.ElementZ [page 250](#)

HasLayerPorosity

[Get/Set] Property HasLayerPorosity[lay : Integer] : Boolean;

Description

Specifies if the layer number *lay* is porous or not.

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Target.NumberOfLayers}$.

Related Properties and Methods

Target.PoreDiameter [page 256](#)
Target.PorosityFraction [page 256](#)

HasLayerRoughness

[Get/Set] Property HasLayerRoughness[lay : Integer] : Boolean;

Description

Specifies if the layer number *lay* is rough or not. The FWHM of the roughness is specified by [Target.LayerRoughness](#). If *HasLayerRoughness* is *false*, the layer is treated as smooth, and [Target.LayerRoughness](#) is ignored.

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Target.NumberOfLayers}$.

Related Properties and Methods

Target.LayerRoughness [page 254](#)

HasSubstrateRoughness

[Get/Set] Property HasSubstrateRoughness : Boolean;

Description

Returns *true* if the substrate is rough, and *false* if substrate roughness is switched off, i.e. the substrate is smooth. Substrate roughness can be switched off by setting *HasSubstrateRoughness* to *false*. If *HasSubstrateRoughness* is set to *true*, Lorentzian substrate roughness is selected. The FWHM of the roughness is specified by [Target.SubstrateRoughness](#). If *HasSubstrateRoughness* is *false* then [Target.SubstrateRoughness](#) is ignored.

Note: This function is obsolete and maintained only for backward compatibility to earlier versions of SIMNRA. Use [Target.SubstrateRoughnessDistribution](#) instead.

Related Properties and Methods

Target.SubstrateRoughness [page 257](#)

Target.SubstrateRoughnessDistribution [page 257](#)

IsotopeConcentration

[Get/Set] Property IsotopeConcentration[lay, el, iso : Integer] : Double;

Description

Concentration of isotope number *iso* in element number *el* in layer number *lay*. The sum of concentrations of all isotopes in an elements must be equal to 1. This is not checked by SIMNRA. If *IsotopeConcentration* is changed, it is the responsibility of the programmer to assure that the sum of the concentrations is equal to 1.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq \text{Target.NumberOfLayers}$.
<i>el</i>	Number of the element, with $1 \leq el \leq \text{Target.NumberOfElements}[lay]$.
<i>iso</i>	Number of the isotope, with $1 \leq iso \leq \text{Target.NumberOfIsotopes}[lay, el]$.

Related Properties and Methods

Target.IsotopeMass [page 254](#)

IsotopeMass

[Get/Set] Property IsotopeMass[lay, el, iso : Integer] : Double;

A. OLE automation reference

Description

Mass of isotope number *iso* in element number *el* in layer number *lay* [amu].

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq Target.NumberOfLayers$.
<i>el</i>	Number of the element, with $1 \leq el \leq Target.NumberOfElements[lay]$.
<i>iso</i>	Number of the isotope, with $1 \leq iso \leq Target.NumberOfIsotopes[lay, el]$.

Related Properties and Methods

Target.IsotopeConcentration [page 253](#)

LayerRoughness

[Get/Set] Property LayerRoughness[lay : Integer] : Double;

Description

FWHM of the roughness of layer number *lay* [10^{15} atoms/cm²]. [Target.HasLayerRoughness\[lay\]](#) must be *true*, otherwise *LayerRoughness* has no effect.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq Target.NumberOfLayers$.
------------	---

Related Properties and Methods

Target.HasLayerRoughness [page 252](#)

LayerThickness

[Get/Set] Property LayerThickness[lay : Integer] : Double;

Description

Thickness of layer number *lay* [10^{15} atoms/cm²].

Parameters

lay Number of the layer, with $1 \leq lay \leq Target.NumberOfLayers$.

Related Properties and Methods

Target.Thickness [page 258](#)

NumberOfElements

[Get] Property NumberOfElements[lay : Integer] : Integer;

Description

Number of different elements in layer number *lay*. *NumberOfElements* is readonly.

Parameters

lay Number of the layer, with $1 \leq lay \leq Target.NumberOfLayers$.

Related Properties and Methods

Target.NumberOfIsotopes [page 255](#)

Target.NumberOfLayers [page 256](#)

Target.TotalNumberOfElements [page 258](#)

NumberOfIsotopes

[Get] Property NumberOfIsotopes[lay, el : Integer] : Integer;

Description

Number of different isotopes in element number *el* in layer number *lay*. *NumberOfIsotopes* is readonly.

Parameters

lay Number of the layer, with $1 \leq lay \leq Target.NumberOfLayers$.

Parameters

el Number of the element, with $1 \leq el \leq Target.NumberOfElements[lay]$.

Related Properties and Methods

Target.NumberOfElements [page 255](#)

Target.NumberOfLayers [page 256](#)

NumberOfLayers

[Get] Property NumberOfLayers : Integer;

Description

Total number of layers in the target. *NumberOfLayers* is readonly.

Related Properties and Methods

Target.NumberOfElements [page 255](#)

Target.NumberOfIsotopes [page 255](#)

PoreDiameter

[Get/Set] Property PoreDiameter[lay : Integer] : Double;

Description

Diameter of pores in layer number *lay*. In 10^{15} atoms/cm².

Parameters

lay Number of the layer, with $1 \leq lay \leq Target.NumberOfLayers$.

Related Properties and Methods

Target.HasLayerPorosity [page 252](#)

Target.PorosityFraction [page 256](#)

PorosityFraction

[Get/Set] Property PorosityFraction[lay : Integer] : Double;

Description

Volume fraction of pores in layer number *lay*, with $0 \leq PorosityFraction \leq 1$.

Parameters

lay Number of the layer, with $1 \leq lay \leq Target.NumberOfLayers$.

Related Properties and Methods

[Target.HasLayerPorosity](#) [page 252](#)

[Target.PoreDiameter](#) [page 256](#)

StoppingFactor

[Get/Set] Property StoppingFactor[lay, Z : Integer] : Double;

Description

Correction factor to the stopping power of a layer, see [section 3.6.1](#).

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq \text{Target.NumberOfLayers}$.
<i>Z</i>	Atomic number of the ion for which the correction factor is valid. Different correction factors can be defined for different ion species.

SubstrateRoughness

[Get/Set] Property SubstrateRoughness : Double;

Description

FWHM of the substrate roughness [deg]. [Target.SubstrateRoughnessDistribution](#) must be $\neq 0$ (or [Target.HasSubstrateRoughness](#) must be *true*), otherwise *SubstrateRoughness* has no effect.

Related Properties and Methods

[Target.HasSubstrateRoughness](#) [page 253](#)

[Target.SubstrateRoughnessDistribution](#) [page 257](#)

SubstrateRoughnessDistribution

[Get/Set] Property SubstrateRoughnessDistribution : Integer;

Description

A. OLE automation reference

Distribution function of the substrate roughness. Allowed values are:

- 0: Smooth substrate. Substrate roughness is switched off. Same as [*Target.HasSubstrateRoughness := false*](#).
- 1: Lorentzian roughness distribution. Same as [*Target.HasSubstrateRoughness := true*](#)
- 2: Gaussian roughness distribution.

The FWHM of the substrate roughness is set with [*Target.SubstrateRoughness*](#).

Related Properties and Methods

[Target.HasSubstrateRoughness](#) [page 253](#)

[Target.SubstrateRoughness](#) [page 257](#)

Thickness

[Get] Property Thickness : Double;

Description

Total thickness of the target in 10^{15} atoms/cm². *Thickness* is read only.

Related Properties and Methods

[Target.LayerThickness](#) [page 254](#)

TotalNumberOfElements

[Get] Property TotalNumberOfElements : Integer;

Description

Total number of different elements in the target. *TotalNumberOfElements* is readonly.

Related Properties and Methods

[Target.NumberOfElements](#) [page 255](#)

A.9.2. Methods

AddElement

Function `AddElement(lay : Integer): Boolean;`

Description

Adds an element to layer number *lay*. The element has no name and zero concentration. After adding, the properties of the element have to be set with [Target.ElementName](#) or [Target.ElementZ](#) and [Target.ElementConcentration](#). If a new element (i.e. an element not already contained in the target) is added, then the cross-section data have to be defined with the methods of the [Simnra.CrossSec](#) object.

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Target.NumberOfLayers}$.

Return Value

Returns *true* if the element was added successfully.

Related Properties and Methods

`Target.AddElementProperties` [page 259](#)

`Target.AddElements` [page 260](#)

`Target.AddIsotope` [page 261](#)

`Target.AddLayer` [page 261](#)

`Target.ElementConcentration` [page 249](#)

`Target.ElementName` [page 250](#)

`Target.ElementZ` [page 250](#)

AddElementProperties

Function `AddElementProperties(lay : Integer; Z : PsafeArray; c : PsafeArray): Boolean;`

Description

Adds multiple elements to layer number *lay* and sets the nuclear charges *Z* and concentrations *c* of these elements. The number of elements is determined from the number of elements in *Z*. The arrays *Z* and *c* must contain the same number of elements.

If a new element (i.e. an element not already contained in the target) is added, then the cross-section data have to be defined with the methods of the [Simnra.CrossSec](#) object.

See [subsection A.18.3](#) for an example how to use *AddElementProperties*.

AddElementProperties is the fastest method for adding multiple elements and setting their properties.

A. OLE automation reference

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq \text{Target.NumberOfLayers}$.
<i>Z</i>	Safe array with nuclear charges of the elements to be added. <i>Z</i> is a safe array of type Integer.
<i>c</i>	Safe array with atomic concentrations of the elements to be added. <i>c</i> is a safe array of type Double.

Return Value

Returns *true* if the elements were added successfully.

Related Properties and Methods

Target.AddElement [page 259](#)
Target.AddElements [page 260](#)
Target.ElementConcentration [page 249](#)
Target.ElementConcentrationArray [page 249](#)
Target.ElementZ [page 250](#)
Target.ElementZArray [page 251](#)

AddElements

Function AddElements(*lay*, NumElements : Integer): Boolean;

Description

Adds *NumElements* elements to layer number *lay*. The elements has no name and zero concentration. After adding, the properties of the elements have to be set with [Target.ElementName](#) or [Target.ElementZ](#) and [Target.ElementConcentration](#). If a new element (i.e. an element not already contained in the target) is added, then the cross-section data have to be defined with the methods of the [Simnra.CrossSec](#) object.

AddElements is faster than successive calls to [Target.AddElement](#) when used for multiple elements.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq \text{Target.NumberOfLayers}$.
<i>Numelements</i>	Number of elements to be added.

Return Value

Returns *true* if the elements were added successfully.

Related Properties and MethodsTarget.AddElement [page 259](#)Target.AddElementProperties [page 259](#)Target.AddLayers [page 262](#)

AddIsotope

Function AddIsotope(lay, el : Integer): Boolean;

Description

Adds an isotope to element number *el* in layer number *lay*. The isotope has no mass and zero concentration. After adding, the properties of the isotope have to be set with [IsotopeMass](#) and [IsotopeConcentration](#).

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq \text{Target.NumberOfLayers}$.
<i>el</i>	Number of the element, with $1 \leq el \leq \text{Target.NumberOfElements}[lay]$.

Return Value

Returns *true* if the isotope was added successfully.

Related Properties and MethodsTarget.AddElement [page 259](#)Target.AddLayer [page 261](#)Target.IsotopeConcentration [page 253](#)Target.IsotopeMass [page 254](#)

AddLayer

Function AddLayer : Boolean;

Description

Adds a new layer. The layer is the last layer in the stack, has zero thickness and contains no elements. After adding a layer at least one element has to be added with [Target.AddElement](#) and layer properties like thickness, roughness etc. have to be set. For adding multiple layers [Target.AddLayers](#) is generally faster.

Parameters

A. OLE automation reference

None.

Return Value

Returns *true* if the layer was added successfully.

Related Properties and Methods

Target.AddElement [page 259](#)

Target.AddIsotope [page 261](#)

Target.AddLayers [page 262](#)

Target.InsertLayer [page 265](#)

AddLayers

Function AddLayers(NumLayers : Integer) : Boolean;

Description

Adds *NumLayers* new layers. The layers are the last layers in the stack, have zero thickness and contain no elements. After adding layers at least one element has to be added to each layer with [Target.AddElement](#) and layer properties like thickness, roughness etc. have to be set.

AddLayers is faster than successive calls to [Target.AddLayer](#) when used for multiple layers.

Parameters

NumLayers Number of layers to be added.

Return Value

Returns *true* if the layers were added successfully.

Related Properties and Methods

Target.AddLayer [page 261](#)

Target.InsertLayer [page 265](#)

Copy

Function Copy(T : ITarget) : Boolean;

Description

Copies all data from *Simnra.Target* object *T* to the current object.
 The following code copies all data from target *Target2* to *Target1*:

```
var
  Target1, Target2 : Variant;
begin
  Target1 := CreateOLEObject(Simnra.Target');
  Target2 := CreateOLEObject(Simnra.Target');
  Target1.Copy(Target2);
end;
```

Parameters

T *Simnra.Target* object.

Return Value

Returns *true* if ok.

Related Properties and Methods

App.Copy [page 207](#)
 Calc.Copy [page 245](#)
 Foil.Copy [page 274](#)
 Projectile.Copy [page 233](#)
 Setup.Copy [page 228](#)
 Spectrum.Copy [page 300](#)
 Window.Copy [page 286](#)

DeleteAllLayer

Function DeleteAllLayer : Boolean;

Description

Deletes all layers from the target.

Parameters

None.

Return Value

Returns *true* if all layers were deleted successfully.

Related Properties and Methods

Target.DeleteLayer [page 265](#)

DeleteElement

Function DeleteElement(*lay*, *el* : Integer) : Boolean;

Description

Deletes element number *el* in layer number *lay*.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq Target.NumberOfLayers$.
<i>el</i>	Number of the element, with $1 \leq el \leq Target.NumberOfElements[lay]$.

Return Value

Returns *true* if the element was deleted successfully.

Related Properties and Methods

Target.DeleteIsotope [page 264](#)

Target.DeleteLayer [page 265](#)

DeleteIsotope

Function DeleteIsotope(*lay*, *el*, *iso* : Integer) : Boolean;

Description

Deletes isotope number *iso* in element number *el* in layer number *lay*.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq Target.NumberOfLayers$.
<i>el</i>	Number of the element, with $1 \leq el \leq Target.NumberOfElements[lay]$.
<i>iso</i>	Number of the isotope, with $1 \leq iso \leq Target.NumberOfIsotopes[lay, el]$.

Return Value

Returns *true* if the element was deleted successfully.

Related Properties and Methods

Target.DeleteElement [page 264](#)

Target.DeleteLayer [page 265](#)

DeleteLayer

Function DeleteLayer(layer : Integer) : Boolean;

Description

Deletes layer number *lay*.

Parameters

<i>lay</i>	Number of the layer to delete, with $1 \leq lay \leq Target.NumberOfLayers$.
------------	---

Return Value

Returns *true* if the layer was deleted successfully.

Related Properties and Methods

Target.DeleteAllLayer [page 263](#)

Target.DeleteElement [page 264](#)

Target.DeleteIsotope [page 264](#)

InsertLayer

Function InsertLayer(layer : Integer) : Boolean;

Description

Inserts a new layer in front of layer number *lay*. The layer has zero thickness and contains no elements. After inserting a layer at least one element has to be added with [Target.AddElement](#) and layer properties like thickness, roughness etc. have to be set.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq Target.NumberOfLayers$.
------------	---

A. OLE automation reference

Return Value

Returns *true* if the layer was inserted successfully.

Related Properties and Methods

Target.AddLayer [page 261](#)

ReadLayer

Procedure ReadLayer(FileName : WideString; lay : Integer)

Description

Reads a layer description from file *FileName* and replaces layer number *lay*. The layer *lay* must exist, but may be empty. Its current content is replaced by the content from *FileName*.

Parameters

FileName Name of the file including path.

lay Number of the layer, with $1 \leq lay \leq \text{Target.NumberOfLayers}$.

Return Value

None.

Related Properties and Methods

Target.SaveLayerAs [page 267](#)

ReadTarget

Procedure ReadTarget(FileName : WideString);

Description

Reads a target description from file *FileName*. The current target is replaced by the content of *FileName*. *FileName* must be a valid xnra file, IBA data format file, nra file of SIMNRA 6 or earlier, or target description file of SIMNRA 6 or earlier. The file type is determined automatically from the file extension.

Parameters

FileName Name of the file including path.

Related Properties and Methods

Target.ReadLayer [page 266](#)

Target.SaveTargetAs [page 267](#)

SaveLayerAs

Procedure SaveLayerAs(FileName : WideString; lay : Integer)

Description

Saves the description of layer *lay* to file *FileName*. The layer *lay* must exist.

Parameters

FileName Name of the file including path.

lay Number of the layer, with $1 \leq lay \leq \text{Target.NumberOfLayers}$.

Return Value

None.

Related Properties and Methods

Target.ReadLayer [page 266](#)

SaveTargetAs

Procedure SaveTargetAs(FileName : WideString);

Description

Saves the target description to file *FileName*. The data are stored in xnra file format.

Parameters

FileName Name of the file including path.

Related Properties and Methods

A. OLE automation reference

Target.ReadTarget [page 266](#)

Target.SaveLayerAs [page 267](#)

A.10. Simnra.Foil

The *Simnra.Foil* object represents the foil in front of the detector with all layers and elements.

A.10.1. Properties

ElementAmount

[Get/Set] Property ElementAmount[lay, el : Integer] : Double;

Description

Amount of element number *el* in layer number *lay* [10^{15} atoms/cm²].

Parameters

<i>lay</i>	Number of the layer, with $1 \leq \textit{lay} \leq \textit{Foil.NumberOfLayers}$.
<i>el</i>	Number of the element, with $1 \leq \textit{el} \leq \textit{Foil.NumberOfElements}[\textit{lay}]$.

Related Properties and Methods

Foil.ElementConcentration [page 268](#)

Foil.LayerThickness [page 271](#)

ElementConcentration

[Get/Set] Property ElementConcentration[lay, el : Integer] : Double;

Description

Concentration of element number *el* in layer number *lay*. The sum of concentrations of all elements in a layer must be equal to 1. This is not checked by SIMNRA. If *ElementConcentration* is changed, it is the responsibility of the programmer to assure that the sum of the concentrations is equal to 1.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq \textit{lay} \leq \textit{Foil.NumberOfLayers}$.
------------	---

el Number of the element, with
 $1 \leq el \leq \text{Foil.NumberOfElements}[\textit{lay}]$.

Related Properties and Methods

Foil.AddElement [page 273](#)

Foil.ElementName [page 269](#)

Foil.ElementZ [page 269](#)

ElementName

[Get/Set] Property ElementName[lay, el : Integer] : WideString;

Description

Name of element number *el* in layer number *lay*. Returns 'XX' if the element is unknown. See [Target.ElementName](#) for a description of the name format.

Parameters

lay Number of the layer, with $1 \leq \textit{lay} \leq \text{Foil.NumberOfLayers}$.

el Number of the element, with
 $1 \leq el \leq \text{Foil.NumberOfElements}[\textit{lay}]$.

Related Properties and Methods

Foil.AddElement [page 273](#)

Foil.ElementConcentration [page 268](#)

Foil.ElementZ [page 269](#)

ElementZ

[Get/Set] Property ElementZ[lay, el : Integer] : Integer;

Description

Nuclear Charge Z of element number *el* in layer number *lay*. Returns -1 if the element is unknown.

Parameters

lay Number of the layer, with $1 \leq \textit{lay} \leq \text{Foil.NumberOfLayers}$.

el Number of the element, with
 $1 \leq el \leq \text{Foil.NumberOfElements}[\textit{lay}]$.

Related Properties and Methods

Foil.AddElement [page 273](#)
Foil.ElementConcentration [page 268](#)
Foil.ElementName [page 269](#)

HasLayerPorosity

[Get/Set] Property HasLayerPorosity[lay : Integer] : Boolean;

Description

Specifies if the layer number *lay* is porous or not.

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Foil.NumberOfLayers}$.

Related Properties and Methods

Foil.PoreDiameter [page 272](#)
Foil.PorosityFraction [page 272](#)

HasLayerRoughness

[Get/Set] Property HasLayerRoughness[lay : Integer] : Boolean;

Description

Specifies if the layer number *lay* is rough or not. The FWHM of the roughness is specified by [Foil.LayerRoughness\[lay\]](#). If *HasLayerRoughness* is *false*, the layer is treated as smooth, and [Foil.LayerRoughness](#) is ignored.

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Foil.NumberOfLayers}$.

Related Properties and Methods

Foil.LayerRoughness [page 271](#)

LayerRoughness

[Get/Set] Property LayerRoughness[lay : Integer] : Double;

Description

FWHM of the roughness of layer number *lay* [10^{15} atoms/cm²]. [Foil.HasLayerRoughness\[lay\]](#) must be *true*, otherwise *LayerRoughness* has no effect.

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Foil.NumberOfLayers}$.

Related Properties and Methods

Foil.HasLayerRoughness [page 270](#)

LayerThickness

[Get/Set] Property LayerThickness[lay : Integer] : Double;

Description

Thickness of layer number *lay* [10^{15} atoms/cm²].

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Foil.NumberOfLayers}$.

Related Properties and Methods

Foil.Thickness [page 273](#)

NumberOfElements

[Get] Property NumberOfElements[lay : Integer] : Integer;

Description

Number of different elements in layer number *lay*. *NumberOfElements* is readonly.

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Foil.NumberOfLayers}$.

NumberOfLayers

[Get] Property NumberOfLayers : Integer;

Description

Total number of layers in the foil. *NumberOfLayers* is readonly.

PoreDiameter

[Get/Set] Property PoreDiameter[lay : Integer] : Double;

Description

Diameter of pores in layer number *lay*. In 10^{15} atoms/cm².

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Foil.NumberOfLayers}$.

Related Properties and Methods

Foil.HasLayerPorosity [page 270](#)

Foil.PorosityFraction [page 272](#)

PorosityFraction

[Get/Set] Property PorosityFraction[lay : Integer] : Double;

Description

Volume fraction of pores in layer number *lay*, with $0 \leq \text{PorosityFraction} \leq 1$.

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Foil.NumberOfLayers}$.

Related Properties and Methods

Foil.HasLayerPorosity [page 270](#)

Foil.PoreDiameter [page 272](#)

StoppingFactor

[Get/Set] Property StoppingFactor[lay, Z : Integer] : Double;

Description

Correction factor to the stopping power of a layer, see [section 3.6.1](#).

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq Foil.NumberOfLayers$.
<i>Z</i>	Atomic number of the ion for which the correction factor is valid. Different correction factors can be defined for different ion species.

Thickness

[Get] Property Thickness : Double;

Description

Total thickness of the foil in 10^{15} atoms/cm². *Thickness* is read only.

Related Properties and Methods

Foil.LayerThickness [page 271](#)

A.10.2. Methods

AddElement

Function AddElement(lay : Integer): Boolean;

Description

Adds an element to layer number *lay*. The element has no name and zero concentration. After adding the elements properties have to be set with [Foil.ElementName](#) and [Foil.ElementConcentration](#).

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq Foil.NumberOfLayers$.
------------	---

A. OLE automation reference

Return Value

Returns *true* if the element was added successfully.

Related Properties and Methods

Foil.AddLayer [page 274](#)

Foil.ElementConcentration [page 268](#)

Foil.ElementName [page 269](#)

Foil.ElementZ [page 269](#)

AddLayer

Function AddLayer : Boolean;

Description

Adds a new layer. The layer is the last layer in the stack, has zero thickness and contains no elements. After adding a layer at least one element has to be added with [Foil.AddElement](#) and layer properties like thickness, roughness etc. have to be set.

Parameters

None.

Return Value

Returns *true* if the layer was added successfully.

Related Properties and Methods

Foil.AddElement [page 273](#)

Foil.InsertLayer [page 277](#)

Copy

Function Copy(F : IFoil) : Boolean;

Description

Copies all data from *Simnra.Foil* object *F* to the current object.

The following code copies all data from foil *Foil2* to *Foil1*:

```
var
  Foil1, Foil2 : Variant;
begin
  Foil1 := CreateOLEObject(Simnra.Foil');
  Foil2 := CreateOLEObject(Simnra.Foil');
  Foil1.Copy(Foil2);
end;
```

Parameters

F *Simnra.Foil* object.

Return Value

Returns *true* if ok.

Related Properties and Methods

App.Copy [page 207](#)

Calc.Copy [page 245](#)

Projectile.Copy [page 233](#)

Setup.Copy [page 228](#)

Spectrum.Copy [page 300](#)

Target.Copy [page 262](#)

Window.Copy [page 286](#)

DeleteAllLayer

Function DeleteAllLayer : Boolean;

Description

Deletes all layers from the foil.

Parameters

None.

Return Value

Returns *true* if all layers were deleted successfully.

Related Properties and Methods

Foil.DeleteLayer [page 276](#)

DeleteElement

Function DeleteElement(*lay*, *el* : Integer) : Boolean;

Description

Deletes element number *el* in layer number *lay*.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq \textit{lay} \leq \textit{Foil.NumberOfLayers}$.
<i>el</i>	Number of the element, with $1 \leq \textit{el} \leq \textit{Foil.NumberOfElements}[\textit{lay}]$.

Return Value

Returns *true* if the element was deleted successfully.

Related Properties and Methods

Foil.DeleteLayer [page 276](#)

DeleteLayer

Function DeleteLayer(*lay* : Integer) : Boolean;

Description

Deletes layer number *lay*.

Parameters

<i>lay</i>	Number of the layer to delete, with $1 \leq \textit{lay} \leq \textit{Foil.NumberOfLayers}$.
------------	---

Return Value

Returns *true* if the layer was deleted successfully.

Related Properties and Methods

Foil.DeleteAllLayer [page 275](#)

Foil.DeleteElement [page 276](#)

InsertLayer

Function InsertLayer(*lay* : Integer) : Boolean;

Description

Inserts a new layer in front of layer number *lay*. The layer has zero thickness and contains no elements. After inserting a layer at least one element has to be added with [AddElement](#) and layer properties like thickness, roughness etc. have to be set.

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Foil.NumberOfLayers}$.

Return Value

Returns *true* if the layer was inserted successfully.

Related Properties and Methods

Foil.AddLayer [page 274](#)

ReadLayer

Procedure ReadLayer(*FileName* : WideString; *lay* : Integer)

Description

Reads a layer description from file *FileName* and replaces layer number *lay*. The layer *lay* must exist, but may be empty. Its current content is replaced by the content from *FileName*.

Parameters

FileName Name of the file including path.

lay Number of the layer, with $1 \leq lay \leq \text{Foil.NumberOfLayers}$.

Return Value

None.

Related Properties and Methods

Foil.SaveLayerAs [page 279](#)

ReadFoil

Procedure ReadFoil(FileName : WideString);

Description

Reads a foil description from file *FileName*. The current foil is replaced by the content of *FileName*. *FileName* must be a valid xnra file, IBA data format file, nra file of SIMNRA 6 or earlier, or target description file of SIMNRA 6 or earlier. The file type is determined automatically from the file extension.

Parameters

<i>FileName</i>	Name of the file including path.
-----------------	----------------------------------

Related Properties and Methods

Foil.ReadLayer [page 277](#)

Foil.SaveFoilAs [page 278](#)

SaveFoilAs

Procedure SaveFoilAs(FileName : WideString);

Description

Saves the foil description to file *FileName*. The data are stored in xnra file format.

Parameters

<i>FileName</i>	Name of the file including path.
-----------------	----------------------------------

Related Properties and Methods

Foil.ReadFoil [page 278](#)

Foil.SaveLayerAs [page 279](#)

SaveLayerAs

Procedure SaveLayerAs(FileName : WideString; lay : Integer)

Description

Saves the description of layer *lay* to file *FileName*. The layer *lay* must exist.

Parameters

<i>FileName</i>	Name of the file including path.
<i>lay</i>	Number of the layer, with $1 \leq \textit{lay} \leq \textit{Foil.NumberOfLayers}$.

Return Value

None.

Related Properties and Methods

Foil.ReadLayer [page 277](#)

A.11. Simnra.Window

The *Simnra.Window* object represents the beam window with all layers and elements.

A.11.1. Properties

ElementAmount

[Get/Set] Property ElementAmount[lay, el : Integer] : Double;

Description

Amount of element number *el* in layer number *lay* [10^{15} atoms/cm²].

Parameters

<i>lay</i>	Number of the layer, with $1 \leq \textit{lay} \leq \textit{Window.NumberOfLayers}$.
<i>el</i>	Number of the element, with $1 \leq \textit{el} \leq \textit{Window.NumberOfElements}[\textit{lay}]$.

Related Properties and Methods

Window.ElementConcentration [page 280](#)

Window.LayerThickness [page 282](#)

ElementConcentration

[Get/Set] Property ElementConcentration[lay, el : Integer] : Double;

Description

Concentration of element number *el* in layer number *lay*. The sum of concentrations of all elements in a layer must be equal to 1. This is not checked by SIMNRA. If *ElementConcentration* is changed, it is the responsibility of the programmer to assure that the sum of the concentrations is equal to 1.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq \text{Window.NumberOfLayers}$.
<i>el</i>	Number of the element, with $1 \leq el \leq \text{Window.NumberOfElements}[lay]$.

Related Properties and Methods

Window.AddElement [page 285](#)
Window.ElementName [page 280](#)
Window.ElementZ [page 281](#)

ElementName

[Get/Set] Property ElementName[lay, el : Integer] : WideString;

Description

Name of element number *el* in layer number *lay*. Returns 'XX' if the element is unknown. See [Target.ElementName](#) for a description of the name format.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq \text{Window.NumberOfLayers}$.
<i>el</i>	Number of the element, with $1 \leq el \leq \text{Window.NumberOfElements}[lay]$.

Related Properties and Methods

Window.AddElement [page 285](#)
Window.ElementConcentration [page 280](#)
Window.ElementZ [page 281](#)

ElementZ

[Get/Set] Property ElementZ[lay, el : Integer] : Integer;

Description

Nuclear Charge Z of element number *el* in layer number *lay*. Returns -1 if the element is unknown.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq \text{Window.NumberOfLayers}$.
<i>el</i>	Number of the element, with $1 \leq el \leq \text{Window.NumberOfElements}[lay]$.

Related Properties and Methods

Window.AddElement [page 285](#)
 Window.ElementConcentration [page 280](#)
 Window.ElementName [page 280](#)

HasLayerPorosity

[Get/Set] Property HasLayerPorosity[lay : Integer] : Boolean;

Description

Specifies if the layer number *lay* is porous or not.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq \text{Window.NumberOfLayers}$.
------------	--

Related Properties and Methods

Window.PoreDiameter [page 283](#)
 Window.PorosityFraction [page 284](#)

HasLayerRoughness

[Get/Set] Property HasLayerRoughness[lay : Integer] : Boolean;

Description

A. OLE automation reference

Specifies if the layer number *lay* is rough or not. The FWHM of the roughness is specified by [Window.LayerRoughness](#). If *HasLayerRoughness* is *false*, the layer is treated as smooth, and [Window.LayerRoughness](#) is ignored.

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Window.NumberOfLayers}$.

Related Properties and Methods

Window.LayerRoughness [page 282](#)

LayerRoughness

[Get/Set] Property LayerRoughness[*lay* : Integer] : Double;

Description

FWHM of the roughness of layer number *lay* [10^{15} atoms/cm²]. Window.HasLayerRoughness[*lay*] must be *true*, otherwise *LayerRoughness* has no effect.

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Window.NumberOfLayers}$.

Related Properties and Methods

Window.HasLayerRoughness [page 281](#)

LayerThickness

[Get/Set] Property LayerThickness[*lay* : Integer] : Double;

Description

Thickness of layer number *lay* [10^{15} atoms/cm²].

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Window.NumberOfLayers}$.

Related Properties and Methods

Foil.Thickness [page 273](#)

NumberOfElements

[Get] Property NumberOfElements[lay : Integer] : Integer;

Description

Number of different elements in layer number *lay*. *NumberOfElements* is readonly.

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Window.NumberOfLayers}$.

NumberOfLayers

[Get] Property NumberOfLayers : Integer;

Description

Total number of layers in the Window. *NumberOfLayers* is readonly.

PoreDiameter

[Get/Set] Property PoreDiameter[lay : Integer] : Double;

Description

Diameter of pores in layer number *lay*. In 10^{15} atoms/cm².

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Window.NumberOfLayers}$.

Related Properties and Methods

Window.HasLayerPorosity [page 281](#)

Window.PorosityFraction [page 284](#)

PorosityFraction

[Get/Set] Property PorosityFraction[lay : Integer] : Double;

A. OLE automation reference

Description

Volume fraction of pores in layer number *lay*, with $0 \leq \text{PorosityFraction} \leq 1$.

Parameters

lay Number of the layer, with $1 \leq \text{lay} \leq \text{Window.NumberOfLayers}$.

Related Properties and Methods

Window.HasLayerPorosity [page 281](#)

Window.PoreDiameter [page 283](#)

StoppingFactor

[Get/Set] Property StoppingFactor[lay, Z : Integer] : Double;

Description

Correction factor to the stopping power of a layer, see [section 3.6.1](#).

Parameters

lay Number of the layer, with $1 \leq \text{lay} \leq \text{Window.NumberOfLayers}$.

Z Atomic number of the ion for which the correction factor is valid. Different correction factors can be defined for different ion species.

Thickness

[Get] Property Thickness : Double;

Description

Total thickness of the window in 10^{15} atoms/cm². *Thickness* is read only.

Related Properties and Methods

Window.LayerThickness [page 282](#)

A.11.2. Methods

AddElement

Function AddElement(lay : Integer): Boolean;

Description

Adds an element to layer number *lay*. The element has no name and zero concentration. After adding the elements properties have to be set with [Window.ElementName](#) and [Window.ElementConcentration](#).

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Window.NumberOfLayers}$.

Return Value

Returns *true* if the element was added successfully.

Related Properties and Methods

Window.AddLayer [page 285](#)

Window.ElementConcentration [page 280](#)

Window.ElementName [page 280](#)

AddLayer

Function AddLayer : Boolean;

Description

Adds a new layer. The layer is the last layer in the stack, has zero thickness and contains no elements. After adding a layer at least one element has to be added with [Window.AddElement](#) and layer properties like thickness, roughness etc. have to be set.

Parameters

None.

Return Value

Returns *true* if the layer was added successfully.

Related Properties and Methods

A. OLE automation reference

Window.AddElement [page 285](#)

Window.InsertLayer [page 288](#)

Copy

Function Copy(W : IWindow) : Boolean;

Description

Copies all data from *Simnra.Window* object *W* to the current window object.

The following code copies all data from window *Window2* to *Window1*:

```
var
  Window1, Window2 : Variant;
begin
  Window1 := CreateOLEObject(Simnra.Window');
  Window2 := CreateOLEObject(Simnra.Window');
  Window1.Copy(Window2);
end;
```

Parameters

<i>W</i>	<i>Simnra.Window</i> object.
----------	------------------------------

Return Value

Returns *true* if ok.

Related Properties and Methods

App.Copy [page 207](#)

Calc.Copy [page 245](#)

Foil.Copy [page 274](#)

Projectile.Copy [page 233](#)

Setup.Copy [page 228](#)

Spectrum.Copy [page 300](#)

Target.Copy [page 262](#)

DeleteAllLayer

Function DeleteAllLayer : Boolean;

Description

Deletes all layers from the window.

Parameters

None.

Return Value

Returns *true* if all layers were deleted successfully.

Related Properties and Methods

Window.DeleteLayer [page 287](#)

DeleteElement

Function DeleteElement(*lay*, *el* : Integer) : Boolean;

Description

Deletes element number *el* in layer number *lay*.

Parameters

<i>lay</i>	Number of the layer, with $1 \leq lay \leq \text{Window.NumberOfLayers}$.
<i>el</i>	Number of the element, with $1 \leq el \leq \text{Window.NumberOfElements}[lay]$.

Return Value

Returns *true* if the element was deleted successfully.

Related Properties and Methods

Window.DeleteLayer [page 287](#)

DeleteLayer

Function DeleteLayer(*lay* : Integer) : Boolean;

Description

Deletes layer number *lay*.

Parameters

A. OLE automation reference

lay Number of the layer to delete, with $1 \leq lay \leq \text{Window.NumberOfLayers}$.

Return Value

Returns *true* if the layer was deleted successfully.

Related Properties and Methods

Window.DeleteAllLayer [page 286](#)

Window.DeleteElement [page 287](#)

InsertLayer

Function InsertLayer(*lay* : Integer) : Boolean;

Description

Inserts a new layer in front of layer number *lay*. The layer has zero thickness and contains no elements. After inserting a layer at least one element has to be added with [Window.AddElement](#) and layer properties like thickness, roughness etc. have to be set.

Parameters

lay Number of the layer, with $1 \leq lay \leq \text{Window.NumberOfLayers}$.

Return Value

Returns *true* if the layer was inserted successfully.

Related Properties and Methods

Window.AddLayer [page 285](#)

ReadLayer

Procedure ReadLayer(*FileName* : WideString; *lay* : Integer)

Description

Reads a layer description from file *FileName* and replaces layer number *lay*. The layer *lay* must exist, but may be empty. Its current content is replaced by the content from *FileName*.

Parameters

<i>FileName</i>	Name of the file including path.
<i>lay</i>	Number of the layer, with $1 \leq lay \leq \text{Window.NumberOfLayers}$.

Return Value

None.

Related Properties and Methods

Window.SaveLayerAs [page 289](#)

ReadWindow

Procedure ReadWindow(FileName : WideString);

Description

Reads a window description from file *FileName*. The current window is replaced by the content of *FileName*. *FileName* must be a valid xnra or IBA data format file. The file type is determined automatically from the file extension.

Parameters

<i>FileName</i>	Name of the file including path.
-----------------	----------------------------------

Related Properties and Methods

Window.ReadLayer [page 288](#)

Window.SaveWindowAs [page 290](#)

SaveLayerAs

Procedure SaveLayerAs(FileName : WideString; lay : Integer)

Description

Saves the description of layer *lay* to file *FileName*. The layer *lay* must exist.

Parameters

<i>FileName</i>	Name of the file including path.
-----------------	----------------------------------

A. OLE automation reference

lay Number of the layer, with $1 \leq lay \leq \text{Window.NumberOfLayers}$.

Return Value

None.

Related Properties and Methods

Window.ReadLayer [page 288](#)

SaveWindowAs

Procedure SaveWindowAs(FileName : WideString);

Description

Saves the window description to file *FileName*. The data are stored in xnra file format.

Parameters

FileName Name of the file including path.

Related Properties and Methods

Window.ReadWindow [page 289](#)

Window.SaveLayerAs [page 289](#)

A.12. Simnra.Fit

The *Simnra.Fit* object represents the fit parameters, i.e. what to fit, number of fit regions, maximum number of iterations etc. The fit parameters in the *Simnra.Fit* object have to be adjusted first before the fit can be performed with the [App.FitSpectrum](#) method, see [page 209](#).

A.12.1. Properties

Accuracy

[Get/Set] Property Accuracy : Double;

Default Value

0.01

Description

Desired accuracy of the fit, see [subsection 3.8.3](#). Fitting will be performed until the desired accuracy is obtained or the maximum number of iterations is reached.

Related Properties and Methods

Fit.MaxIterations [page 293](#)

Chi2Evaluation

[Get/Set] Property Chi2Evaluation: Integer;

Default Value

0

Description

Determines how χ^2 is determined, see [page 49](#) for details. Allowed values are:

0: Channels.

1: Integrals.

EnergyCalibration

[Get/Set] Property EnergyCalibration : Boolean;

Default Value

false

Description

Specifies if the energy calibration (offset and linear term) is fitted. The energy calibration is fitted, if *EnergyCalibration* is *true*.

LayerComposition

[Get/Set] Property LayerComposition : Boolean;

Default Value

false

A. OLE automation reference

Description

Specifies if the composition and thickness of a target layer is fitted. The number of the layer is specified by the [Fit.LayerNr](#) property. The composition of layer number [Fit.LayerNr](#) is fitted, if [Fit.LayerComposition](#) is *true*. Only one layer can be fitted at the same time.

Related Properties and Methods

[Fit.LayerNr](#) [page 292](#)

[Fit.LayerThickness](#) [page 293](#)

LayerNr

[Get/Set] Property LayerNr : Integer;

Default Value

1

Description

Specifies the number of the target layer, which thickness or composition is fitted. Only one layer can be fitted at the same time. [Fit.LayerComposition](#) and/or [Fit.LayerThickness](#) must be *true*, otherwise *LayerNr* has no effect.

Related Properties and Methods

[Fit.LayerComposition](#) [page 291](#)

[Fit.LayerThickness](#) [page 293](#)

LayerRoughness

[Get/Set] Property LayerRoughness : Boolean;

Default Value

false

Description

Specifies if the roughness of a target layer is fitted. The number of the layer is specified by the [Fit.LayerNr](#) property. The roughness of layer number [Fit.LayerNr](#) is fitted, if *LayerRoughness* is *true*. Only one layer can be fitted at the same time.

Related Properties and Methods

[Fit.LayerNr page 292](#)

[Fit.LayerComposition page 291](#)

LayerThickness

[Get/Set] Property LayerThickness : Boolean;

Default Value

false

Description

Identical to [Fit.LayerComposition](#), see [page 291](#). This property is depreciated and only available for backward compatibility with previous versions. This method should not be used for new developments.

Related Properties and Methods

[Fit.LayerNr page 292](#)

[Fit.LayerComposition page 291](#)

MaxIterations

[Get/Set] Property MaxIterations : Integer;

Default Value

20

Description

Maximum number of fit iterations. Fitting will be performed until the desired accuracy is obtained or the maximum number of iterations is reached.

Related Properties and Methods

[Fit.Accuracy page 290](#)

NumberOfRegions

[Get/Set] Property NumberOfRegions : Integer;

A. OLE automation reference

Default Value

1

Description

Number of different regions where χ^2 is calculated, see [subsection 3.8.3](#). At least one region must exist, and the regions should not overlap. The lower and upper channels of the regions are specified by the [Fit.RegionMinChannel](#) and [Fit.RegionMaxChannel](#) properties.

Related Properties and Methods

[Fit.RegionMaxChannel](#) [page 294](#)

[Fit.RegionMinChannel](#) [page 295](#)

ParticlesSr

[Get/Set] Property ParticlesSr : Boolean;

Default Value

false

Description

Specifies if the number of incident particles times solid angle is fitted. The number of incident particles times solid angle is fitted, if *ParticlesSr* is *true*.

RegionMaxChannel

[Get/Set] Property RegionMaxChannel[reg : Integer] : Integer;

Default Value

8192

Description

Upper channel of fit region number *reg*.

Related Properties and Methods

[Fit.NumberOfRegions](#) [page 294](#)

[Fit.RegionMinChannel](#) [page 295](#)

RegionMinChannel

[Get/Set] Property RegionMinChannel[reg : Integer] : Integer;

Default Value

1

Description

Lower channel of fit region number *reg*.

Related Properties and Methods

Fit.NumberOfRegions [page 294](#)

Fit.RegionMaxChannel [page 294](#)

A.12.2. Methods

Chi2

Function Chi2 : Double;

Description

Quadratic deviation χ^2 between the simulated and measured data points, see [subsection 3.8.3](#). χ^2 is weighted with the statistical error of the experimental data. χ^2 is calculated only in the fit regions defined by [Fit.NumberOfRegions](#), [Fit.RegionMaxChannel](#) and [Fit.RegionMinChannel](#). *Chi2* can be used to develop your own fit algorithms.

Parameters

None.

Return Value

Returns χ^2 .

Related Properties and Methods

Fit.NumberOfRegion [page 294](#)

Fit.RegionMaxChannel [page 294](#)

Fit.RegionMinChannel [page 295](#)

A.13. Simnra.Spectrum

The *Simnra.Spectrum* object represents experimental and simulated spectra, and allows to change plot parameters. The *Simnra.Spectrum* object is meaningless if SIMNRA is running in *PIGE* mode, see [section 3.13](#).

A.13.1. Input parameter

The properties and methods of *Simnra.Spectrum* require the input parameter *spID* specifying which spectrum (experimental spectrum or one of the simulated spectra) is accessed. Possible values for *spID* are:

<i>spID</i>	Selected spectrum
1	Experimental data
2	Simulated data (total)
3	Smoothed experimental data
4	Pile-up contribution to the simulated spectrum
5	Dual scattering background (only available after calling App.CalculateDualScatteringBackground)
≥ 6	Simulated data (elements and isotopes)

Other values are not allowed.

spID values for spectra of elements and isotopes depend on the elements and isotopes present in the sample. Use the functions [Spectrum.IDOfElement](#) and [Spectrum.IDOfIsotope](#) to retrieve the *spID* of spectra of specific elements and isotopes.

A.13.2. Properties

AutoScale

[Get/Set] Property AutoScale : Boolean;

Default Value

true

Description

If *AutoScale* is *true* the plot is scaled automatically, if experimental data are imported or a new calculation is performed. If *false*, the axis scales remain fixed.

BottomAxisMax

[Get/Set] Property BottomAxisMax : Double;

Description

Bottom axis maximum.

Related Properties and Methods

BottomAxisMin [page 297](#)

BottomAxisMin

[Get/Set] Property BottomAxisMin : Double;

Description

Bottom axis minimum.

Related Properties and Methods

BottomAxisMax [page 297](#)

EndAcquisition

[Get/Set] Property EndAcquisition : TDateTime;

Description

Ending date and time of the acquisition of the experimental spectrum. *EndAcquisition* is readonly. *EndAcquisition* returns 0.0 if the date and time are unknown.

Related Properties and Methods

StartAcquisition [page 299](#)

LeftAxisMax

[Get/Set] Property LeftAxisMax : Double;

Description

Left axis maximum.

Related Properties and Methods

LeftAxisMin [page 298](#)

LeftAxisMin

[Get/Set] Property LeftAxisMin : Double;

Description

Left axis minimum.

Related Properties and Methods

LeftAxisMax [page 297](#)

M

[Get/Set] Property M[spID : Integer] : Double;

Description

Mass of the isotope of a simulated spectrum of an isotope [amu]. *M* is equal to zero for the total simulated spectrum, spectra of elements, the pile-up spectrum, the dual-scattering background spectrum, and for experimental spectra.

Parameters

spID Spectrum ID, see [page 296](#).

Related Properties and Methods

Spectrum.Z [page 299](#)

spID [page 296](#)

NumberOfChannels

[Get] Property NumberOfChannels[spID : Integer] : Integer;

Description

Number of channels in the spectrum. *NumberOfChannels* is readonly.

Parameters

[spID](#) Spectrum ID, see [page 296](#).

Related Properties and Methods

[spID](#) [page 296](#)

StartAcquisition

[Get/Set] Property StartAcquisition : TDateTime;

Description

Starting date and time of the acquisition of the experimental spectrum. *StartAcquisition* is readonly. *StartAcquisition* returns 0.0 if the date and time are unknown.

Related Properties and Methods

EndAcquisition [page 297](#)

Title

[Get/Set] Property Title[spID : Integer] : WideString;

Description

Title of a spectrum.

Parameters

[spID](#) Spectrum ID, see [page 296](#).

Related Properties and Methods

[spID](#) [page 296](#)

Z

[Get/Set] Property Z[spID : Integer] : Integer;

Description

A. OLE automation reference

Nuclear charge of the element or isotope of a simulated spectrum of an element or isotope. *Z* is equal to zero for the total simulated spectrum, the pile-up spectrum, the dual-scattering background spectrum, and for experimental spectra.

Parameters

[spID](#) Spectrum ID, see [page 296](#).

Related Properties and Methods

[Spectrum.M](#) [page 298](#)

[spID](#) [page 296](#)

A.13.3. Methods

AddSimulated

Function `AddSimulated(S : ISpectrum) : Boolean;`

Description

Adds all simulated spectra from *Simnra.Spectrum* object *S* to the current *Spectrum* object. i.e. the contents of all channels of all simulated spectra are added.

If [Spectrum.AutoScale](#) is *true*, then the plot is scaled automatically to the added simulated spectra.

Parameters

S *Simnra.Spectrum* object.

Return Value

Returns *true* if ok.

Related Properties and Methods

[Spectrum.CopySimulated](#) [page 302](#)

Copy

Function `Copy(S : ISpectrum) : Boolean;`

Description

Copies all spectra (experimental and simulated) from *Simnra.Spectrum* object *S* to the current *Spectrum* object.

The following code copies all spectra from spectrum *Spectrum2* to *Spectrum1*:

```
var
  Spectrum1, Spectrum2 : Variant;
begin
  Spectrum1 := CreateOLEObject(Simnra.Spectrum');
  Spectrum2 := CreateOLEObject(Simnra.Spectrum');
  Spectrum1.Copy(Spectrum2);
end;
```

If *Setup.AutoScale* is *true*, then the plot is scaled automatically.

Copy copies only the data of the spectra, but not the visual appearance in the plot.

Parameters

S *Simnra.Spectrum* object.

Return Value

Returns *true* if ok.

Related Properties and Methods

App.Copy [page 207](#)
 Calc.Copy [page 245](#)
 Foil.Copy [page 274](#)
 Projectile.Copy [page 233](#)
 Setup.Copy [page 228](#)
 Target.Copy [page 262](#)
 Window.Copy [page 286](#)
 Spectrum.CopyExperimental [page 301](#)
 Spectrum.CopySimulated [page 302](#)
 Spectrum.SetDataArray [page 308](#)

CopyExperimental

Function CopyExperimental(*S* : ISpectrum) : Boolean;

Description

A. OLE automation reference

Copies all experimental spectra from *Simnra.Spectrum* object *S* to the current Spectrum object. See [Spectrum.Copy](#) for an example how to use this method.

If [Spectrum.AutoScale](#) is *true*, then the plot is scaled automatically to the experimental spectra.

CopyExperimental copies only the data of the spectra, but not the visual appearance in the plot.

Parameters

S *Simnra.Spectrum* object.

Return Value

Returns *true* if ok.

Related Properties and Methods

[Spectrum.Copy](#) [page 300](#)

[Spectrum.CopySimulated](#) [page 302](#)

[Spectrum.SetDataArray](#) [page 308](#)

CopySimulated

Function CopySimulated(*S* : ISpectrum) : Boolean;

Description

Copies all simulated spectra from *Simnra.Spectrum* object *S* to the current Spectrum object. See [Spectrum.Copy](#) for an example how to use this method.

If [Spectrum.AutoScale](#) is *true*, then the plot is scaled automatically to the simulated spectra.

CopySimulated copies only the data of the spectra, but not the visual appearance in the plot.

Parameters

S *Simnra.Spectrum* object.

Return Value

Returns *true* if ok.

Related Properties and Methods

[Spectrum.AddSimulated](#) [page 300](#)
[Spectrum.Copy](#) [page 300](#)
[Spectrum.CopyExperimental](#) [page 301](#)
[Spectrum.SetDataArray](#) [page 308](#)

Data

Function `Data(spID, chan : Integer) : Double;`

Description

Value of experimental or simulated data in channel *chan*.

Data is intended to access data in one (or a few) channels. [Spectrum.GetDataArray](#) is considerably faster than *Data*, if the whole spectrum is accessed.

Parameters

<i>spID</i>	Spectrum ID, see page 296 .
<i>chan</i>	Channel number, with $0 \leq \textit{chan} \leq \text{Spectrum.NumberOfChannels}[\textit{spID}]$.

Return Value

Returns the value of spectrum *spID* in channel *chan*.

Related Properties and Methods

[Spectrum.DataArray](#) [page 303](#)
[spID](#) [page 296](#)

DataArray

Function `DataArray(spID : Integer) : Variant;`

Description

Returns a whole experimental or simulated spectrum as a SafeArray data structure. Each element in the SafeArray is of type Double. The SafeArray is wrapped into a Variant array for compatibility with OLE automation. See [section A.18](#) for an example how to access the data in the array.

DataArray is considerably faster than [Spectrum.Data](#), if the whole spectrum is accessed.

Note: *DataArray* is depreciated. Use [Spectrum.GetDataArray](#) instead.

Parameters

[spID](#) Spectrum ID, see [page 296](#).

Return Value

Returns a SafeArray with an experimental or simulated spectrum.

Related Properties and Methods

[Spectrum.Data](#) [page 303](#)
[Spectrum.GetDataArray](#) [page 304](#)
[Spectrum.SetDataArray](#) [page 308](#)
[spID](#) [page 296](#)

GetDataArray

Function GetDataArray(spID : Integer) : Variant;

Description

Returns a whole experimental or simulated spectrum as a SafeArray data structure. Each element in the SafeArray is of type Double. The SafeArray is wrapped into a Variant array for compatibility with OLE automation. See [section A.18](#) for an example how to access the data in the array.

GetDataArray is considerably faster than *Spectrum.Data*, if the whole spectrum is accessed.

Parameters

[spID](#) Spectrum ID, see [page 296](#).

Return Value

Returns a SafeArray data structure with an experimental or simulated spectrum wrapped in a variant.

Related Properties and Methods

[Spectrum.Data](#) [page 303](#)
[Spectrum.DataArray](#) [page 303](#)
[Spectrum.SetDataArray](#) [page 308](#)
[spID](#) [page 296](#)

Delete

Procedure Delete(spID : Integer);

Description

Deletes the spectrum *spID*.

If *spID* = 1 (experimental data), then the spectrum with experimental data and all associated spectra (such as smoothed experimental data) are deleted.

If *spID* = 2 (simulated data), then all spectra with simulated data (total spectrum, individual spectra of elements and isotopes, pile-up spectrum, etc.) are deleted.

In all other cases only the selected spectrum is deleted.

Parameters

spID Spectrum ID, see [page 296](#).

Related Properties and Methods

Spectrum.Data [page 303](#)

Spectrum.DataArray [page 303](#)

Spectrum.GetDataArray [page 304](#)

Spectrum.SetDataArray [page 308](#)

spID [page 296](#)

HighElementID

Function HighElementID : Integer;

Description

Returns the spectrum ID *spID* of the last element or isotope spectrum. Together with [Spectrum.LowElementID](#) this allows looping through all simulated spectra of elements and isotopes.

Parameters

None.

Return Value

Returns the *spID* of the last element or isotope spectrum.

Related Properties and Methods

Spectrum.IDOfElement [page 306](#)

Spectrum.IDOfIsotope [page 306](#)

Spectrum.LowElementID [page 307](#)

spID [page 296](#)

IDOfElement

Function IDOfElement(*Z* : Integer) : Integer;

Description

Returns the *spID* of an element spectrum with atomic number *Z*. Returns 0 if a spectrum for that element does not exist.

Parameters

<i>Z</i>	Atomic number of the element.
----------	-------------------------------

Return Value

Returns the *spID* of the spectrum for element with atomic charge *Z*.

Related Properties and Methods

Spectrum.IDOfIsotope [page 306](#)

spID [page 296](#)

IDOfIsotope

Function IDOfIsotope(*Z* : Integer; *M* : Double) : Integer;

Description

Returns the *spID* of the isotope spectrum with atomic number *Z* and mass *M*. Returns 0 if a spectrum for that isotope could not be found.

Parameters

<i>Z</i>	Atomic number of the isotope.
<i>M</i>	Mass of the isotope [amu]. The mass is rounded to the nearest integer value.

Return Value

Returns the *spID* of the spectrum for the isotope with atomic number *Z* and mass *M*.

Related Properties and Methods

Spectrum.IDOfElement [page 306](#)

spID [page 296](#)

Integrate

Function `Integrate(spID, lowChannel, upChannel : Integer) : Double;`

Description

Sum of counts of the experimental or simulated spectrum in the range from *lowChannel* to *upChannel*.

Parameters

<i>spID</i>	Spectrum ID, see page 296 .
<i>lowChannel</i>	Lower channel, with $0 \leq \text{lowChannel} \leq \text{Spectrum.NumberOfChannels}[\text{spID}]$.
<i>upChannel</i>	Upper channel, with $0 \leq \text{upChannel} \leq \text{Spectrum.NumberOfChannels}[\text{spID}]$.

Return Value

Returns the sum of counts.

Related Properties and Methods

spID [page 296](#)

LowElementID

Function `LowElementID : Integer;`

Description

Returns the spectrum ID *spID* of the first element or isotope spectrum. Together with *Spectrum.HighElementID* this allows looping through all simulated spectra of elements and isotopes.

Parameters

None.

Return Value

Returns the *spID* of the first element or isotope spectrum.

Related Properties and Methods

Spectrum.HighElementID [page 305](#)
Spectrum.IDOfElement [page 306](#)
Spectrum.IDOfIsotope [page 306](#)
spID [page 296](#)

SetdataArray

Function SetdataArray(spID : Integer; V : Variant) : Boolean;

Description

Sets an experimental or simulated spectrum identical to the values of a SafeArray data structure. Each element in the SafeArray is of type Double. The SafeArray is wrapped into a Variant array for compatibility with OLE automation. See [section A.18](#) for an example how to access the data in the array. *spID* determines which spectrum is used.

Parameters

<i>spID</i>	Spectrum ID, see page 296 .
<i>V</i>	SafeArray data structure with an experimental or simulated spectrum wrapped in a variant.

Return Value

Returns *true* if ok.

Related Properties and Methods

Spectrum.Copy [page 300](#)
Spectrum.GetDataArray [page 304](#)
spID [page 296](#)

A.14. Simnra.PIGE

The *Simnra.PIGE* object represents energies and yields of Gamma-ray lines. The *Simnra.PIGE* object is meaningless if SIMNRA is running in NRA mode, see [section 3.13](#).

A.14.1. Properties

NumberOfGammas

[Get] Property NumberOfGammas : Integer;

Description

Number of Gamma-ray lines. *NumberOfGammas* = 0 if SIMNRA is running in *NRA* mode. *NumberOfGammas* is readonly.

A.14.2. Methods

Counts

Function Counts(*n* : Integer) : Double;

Description

Number of counts (i.e. total yield) of Gamma-ray line *n*.

Parameters

n Number of the Gamma-ray line, with
 $1 \leq n \leq \text{PIGE.NumberOfGammas}$.

Return Value

Returns the number of counts of Gamma-ray line *n*.

Related Properties and Methods

PIGE.NumberOfGammas [page 309](#)

PIGE.E [page 309](#)

E

Function E(*n* : Integer) : Double;

Description

Energy in keV of Gamma-ray line *n*.

Parameters

n Number of the Gamma-ray line, with
 $1 \leq n \leq \text{PIGE.NumberOfGammas}$.

A. OLE automation reference

Return Value

Returns the energy of Gamma-ray line n in keV.

Related Properties and Methods

PIGE.NumberOfGammas [page 309](#)

PIGE.Counts [page 309](#)

A.15. Simnra.Stopping

The *Simnra.Stopping* object allows to calculate stopping powers, energy losses and energy loss straggling in elements, target layers, foil layers, and window layers.

A.15.1. Input parameter

The methods of *Simnra.Stopping* require the input parameter *TargetID*, which specifies if a layer is in the target, foil or window. The possible values for *TargetID* are:

<i>TargetID</i>	
1	Target
2	Foil
3	Window

Other values are not allowed.

A.15.2. Methods

ClearCache

Procedure ClearCache;

Description

SIMNRA stores necessary stopping power data in an internal memory cache for faster calculation. This procedure clears all cached stopping power data from this cache. The cache is rebuild by reading all necessary stopping power data from disk when the next simulation is performed.

Usually there is no need to call this procedure, as the cache manages itself automatically. You may want to clear the cache and enforce its rebuild if stopping power data files are modified manually.

Parameters

None.

Return Value

None.

Related Properties and MethodsCrossSec.ClearCache [page 318](#)

EnergylossInLayerFunction EnergylossInLayer(*Z1* : Integer; *M1* : Double; *E* : Double; *TargetID* : Integer; *lay* : Integer) : Double;**Description**

Energy loss of an ion *Z1* in a target, foil or window layer. The layer is traversed perpendicularly, incident angle α and exit angle β are ignored. The stopping power model is selected with [Calc.ZBStopping](#) and [Calc.HighEnergyStopping](#) and may be additionally modified by a correction factor to the stopping power, see [section 3.6](#). The accuracy of the energy loss calculation is influenced by the settings of [Calc.AutoStepwidthOut](#) and [Calc.dEOut](#).

Parameters

<i>Z1</i>	Nuclear charge of the ion.
<i>M1</i>	Mass of the ion [amu].
<i>E</i>	Incident energy [keV].
<i>TargetID</i>	Selects target, foil or window, see subsection A.15.1 .
<i>lay</i>	Number of the layer.

Return Value

Returns the energy loss in the layer [keV].

Related Properties and MethodsTargetID [page 310](#)Stopping.StoppingInLayer [page 312](#)

StoppingInElement

A. OLE automation reference

Function StoppingInElement(*Z1* : Integer; *M1* : Double; *E* : Double; *Z2* : Integer) : Double;

Description

Stopping power of an ion *Z1* in element *Z2*. The stopping power model is selected with [Calc.ZBStopping](#) and [Calc.HighEnergyStopping](#).

Parameters

<i>Z1</i>	Nuclear charge of the ion.
<i>M1</i>	Mass of the ion [amu].
<i>E</i>	Incident energy [keV].
<i>Z2</i>	Nuclear charge of the target element.

Return Value

Returns the stopping power in element *Z2* [keV/10¹⁵ atoms/cm²].

Related Properties and Methods

Calc.ZBStopping [page 244](#)

Calc.HighEnergyStopping [page 239](#)

StoppingInLayer

Function StoppingInLayer(*Z1* : Integer; *M1* : Double; *E* : Double; *TargetID* : Integer; *lay* : Integer) : Double;

Description

Stopping power of an ion *Z1* in a target, foil or window layer. The stopping power model is selected with [Calc.ZBStopping](#) and [Calc.HighEnergyStopping](#) and may be additionally modified by a correction factor to the stopping power, see [section 3.6](#).

Parameters

<i>Z1</i>	Nuclear charge of the ion.
<i>M1</i>	Mass of the ion [amu].
<i>E</i>	Incident energy [keV].

<i>TargetID</i>	Selects target, foil or window, see subsection A.15.1 .
<i>lay</i>	Number of the layer.

Return Value

Returns the stopping power in the layer [keV/10¹⁵ atoms/cm²].

Related Properties and Methods

TargetID [page 310](#)

Stopping.EnergyLossInLayer [page 311](#)

StragglingInLayer

Function StragglingInLayer(*Z1* : Integer; *M1* : Double; *E* : Double; *TargetID* : Integer; *lay* : Integer) : Double;

Description

Energy loss straggling of an ion *Z1* in a target, foil or window layer. The layer is traversed perpendicularly, incident angle α and exit angle β are ignored. The electronic energy-loss straggling model is selected with [Calc.StragglingModel](#). See [Stopping.EnergylossInLayer](#) for a list of additional switches, which influence the straggling calculation.

Parameters

<i>Z1</i>	Nuclear charge of the ion.
<i>M1</i>	Mass of the ion [amu].
<i>E</i>	Incident energy [keV].
<i>TargetID</i>	Selects target, foil or window, see subsection A.15.1 .
<i>lay</i>	Number of the layer.

Return Value

Returns the full width at half maximum (FWHM) of the energy loss straggling in the layer [keV].

Related Properties and Methods

A. OLE automation reference

TargetID [page 310](#)

Stopping.EnergylossInLayer [page 311](#)

Calc.StragglngModel [page 243](#)

A.16. Simnra.CrossSec

The *Simnra.CrossSec* object allows to obtain information about available cross-section data sets and selects cross-section data sets for simulation calculations. It returns also Rutherford cross-sections for scattering and recoils.

A.16.1. Properties

Choose

[Get/Set] Property Choose[Index : Integer] : Boolean;

Description

Choose is *true*, if the cross-section data set with index number *Index* is chosen for simulation calculations. Set *Choose* to *true*, if that cross-section data set should be included in simulation calculations.

Warning: *Choose* does not check for overlaps of cross-section data: If you select for example a Rutherford cross-section and a non-Rutherford cross-section data set for the same reaction, then both will be used for the simulation calculation, resulting in incorrect results. SIMNRA does not check for the consistency of cross-section data sets, if these are chosen using the *Simnra.CrossSec* object.

Parameters

<i>Index</i>	Index number of the cross-section data set. $1 \leq \text{Index} \leq \text{CrossSec.Count}$.
--------------	--

Related Properties and Methods

CrossSec.Count [page 314](#)

Count

[Get] Property Count;

Description

Returns the number of available cross-section data sets for all isotopes of all elements in the target. A cross-section data set can be a Rutherford cross-section, a SigmaCalc cross-section, or a cross-section data file containing non-Rutherford scattering or nuclear reaction cross-section data. *Count* is readonly.

Parameters

None.

Return Value

Number of available cross-section data sets.

EMax

[Get/Set] Property EMax[Index : Integer] : Double

Description

Maximum energy until which a cross-section data set is used [keV].

Parameters

Index Index number of the cross-section data set. $1 \leq \text{Index} \leq \text{CrossSec.Count}$.

Related Properties and Methods

CrossSec.Count [page 314](#)

CrossSec.EMin [page 315](#)

EMin

[Get/Set] Property EMin[Index : Integer] : Double

Description

Minimum energy until which a cross-section data set is used [keV].

Parameters

Index Index number of the cross-section data set. $1 \leq \text{Index} \leq \text{CrossSec.Count}$.

Related Properties and Methods

A. OLE automation reference

CrossSec.Count [page 314](#)

CrossSec.EMax [page 315](#)

FileName

[Get] Property FileName[Index : Integer] : WideString;

Description

Returns the file name of the cross-section data set with index *Index*. Rutherford cross-sections are 'Internal'. The file name contains the relative path to the cross-section base directory.

Parameters

<i>Index</i>	Index number of the cross-section data set. $1 \leq \text{Index} \leq \text{CrossSec.Count}$.
--------------	--

Return Value

File name of the cross-section data set with index *Index*.

Related Properties and Methods

CrossSec.Info [page 316](#)

CrossSec.Title [page 317](#)

Info

[Get] Property Info[Index : Integer] : WideString;

Description

Returns an informative string about the cross-section data set with index *Index*.

Parameters

<i>Index</i>	Index number of the cross-section data set. $1 \leq \text{Index} \leq \text{CrossSec.Count}$.
--------------	--

Return Value

Informative string about the cross-section data set with index *Index*.

Related Properties and Methods

CrossSec.FileName [page 316](#)

CrossSec.Title [page 317](#)

ReactionsChosen

[Get/Set] Property ReactionsChosen : Boolean;

Description

Indicates if cross-section data are chosen or not. If *ReactionsChosen* is *false* then the *Reactions* window for choosing cross-section data is shown before a calculation is performed. If *ReactionsChosen* is *true* then the *Reactions* window for choosing cross-section data is not shown.

Related Properties and Methods

CrossSec.SelectReactions [page 320](#)

Title

[Get] Property Title[Index : Integer] : WideString;

Description

Returns the title of the cross-section data set with index *Index*.

Parameters

<i>Index</i>	Index number of the cross-section data set. $1 \leq \text{Index} \leq$ CrossSec.Count .
--------------	--

Return Value

Title of the cross-section data set with index *Index*.

Related Properties and Methods

CrossSec.FileName [page 316](#)

CrossSec.Info [page 316](#)

A.16.2. Methods

ClearCache

Procedure ClearCache;

Description

SIMNRA stores necessary cross-section data in an internal memory cache for faster calculation. This procedure clears all cached cross-section data from this cache. After calling this procedure all cross-sections are undefined.

Usually there is no need to call this procedure, as the cache manages itself automatically. You may want to clear the cache and enforce its rebuild if cross-section data files are modified manually.

Parameters

None.

Return Value

None.

Related Properties and Methods

Stopping.ClearCache [page 310](#)

ERDRutherford

Function ERDRutherford(Z1 : Integer; M1 : Double; Z2 : Integer; M2 : Double; E : Double; Theta : Double) : Double;

Description

Returns the Rutherford recoil cross section for an incident ion $Z1$, $M1$ and a target atom $Z2$, $M2$ at incident energy E and recoil angle Θ . Electronic screening is taken into account, the electronic screening model is selected by [Calc.ScreeningModel](#).

Parameters

$Z1$	Nuclear charge of the incident ion.
$M1$	Mass of the ion [amu].
$Z2$	Nuclear charge of the target atom.

<i>M2</i>	Mass of the target atom [amu].
<i>E</i>	Incident energy [keV].
<i>Theta</i>	Scattering angle [deg].

Return Value

Returns the differential Rutherford cross-section for recoils in the laboratory system [mb/sr].

Related Properties and Methods

CrossSec.RBSRutherford [page 319](#)

Calc.ScreeningModel [page 242](#)

RBSRutherford

Function RBSRutherford(*Z1* : Integer; *M1* : Double; *Z2* : Integer; *M2* : Double; *E* : Double; *Theta* : Double) : Double;

Description

Returns the Rutherford scattering cross section for an incident ion *Z1*, *M1* from a target atom *Z2*, *M2* at incident energy *E* and scattering angle *Theta*. Electronic screening is taken into account, the electronic screening model is selected by [Calc.ScreeningModel](#).

Parameters

<i>Z1</i>	Nuclear charge of the incident ion.
<i>M1</i>	Mass of the ion [amu].
<i>Z2</i>	Nuclear charge of the target atom.
<i>M2</i>	Mass of the target atom [amu].
<i>E</i>	Incident energy [keV].
<i>Theta</i>	Scattering angle [deg].

Return Value

Returns the differential Rutherford cross-section in the laboratory system [mb/sr].

Related Properties and Methods

CrossSec.ERDRutherford [page 318](#)

Calc.ScreeningModel [page 242](#)

SelectReactions

Procedure SelectReactions;

Description

Shows the *Reactions* window for manually selecting cross-section data.

Parameters

None.

Return Value

None.

Related Properties and Methods

CrossSec.ReactionsChosen [page 317](#)

SelectRutherford

Procedure SelectRutherford(Z : Integer);

Description

Selects Rutherford cross-sections for all isotopes of element Z. Rutherford cross-sections for scattering (including low energy solution, if kinematically possible) and recoils (if kinematically possible) are selected. All non-Rutherford and nuclear reaction cross-sections for these isotopes are unselected.

Parameters

Z	Nuclear charge of the element.
---	--------------------------------

Return Value

None.

Related Properties and MethodsCrossSec.SelectRutherfordAll [page 321](#)CrossSec.UnselectRutherford [page 324](#)

SelectRutherfordAll

Procedure SelectRutherfordAll;

Description

Selects Rutherford cross-sections for all isotopes of all elements in the target. Rutherford cross-sections for scattering (including low energy solution, if kinematically possible) and recoils (if kinematically possible) are selected. All non-Rutherford and nuclear reaction cross-sections are unselected.

Parameters

None.

Return Value

None.

Related Properties and MethodsCrossSec.SelectRutherford [page 320](#)CrossSec.UnselectRutherfordAll [page 324](#)

SelectSigmaCalc

Procedure SelectSigmaCalc(Z : Integer);

Description

Selects SigmaCalc cross-sections for all isotopes of element Z , if available. The corresponding Rutherford cross-section is selected for energies below the lowest SigmaCalc energy.

Parameters

Z	Nuclear charge of the element.
-----	--------------------------------

Return Value

None.

Related Properties and Methods

CrossSec.SelectSigmaCalcAll [page 322](#)

CrossSec.UnselectSigmaCalc [page 325](#)

SelectSigmaCalcAll

Procedure SelectSigmaCalcAll;

Description

Selects SigmaCalc cross-sections for all isotopes of all elements in the target, if available. The corresponding Rutherford cross-sections are selected for energies below the lowest SigmaCalc energies.

Parameters

None.

Return Value

None.

Related Properties and Methods

CrossSec.SelectSigmaCalc [page 321](#)

CrossSec.UnselectSigmaCalcAll [page 325](#)

UniversalEMax

Function UniversalEMax(Z1 : Integer; M1 : Double; Z2 : Integer; M2 : Double) : Double;

Description

Returns the maximum energy until which the Universal screening model is used. At higher energies than *UniversalEMax* the Universal screening model is automatically replaced by the Andersen screening model. The Universal screening model is only used, if *Calc.ScreeningModel* is set accordingly, see [page 242](#).

Parameters

<i>Z1</i>	Nuclear charge of the incident ion.
-----------	-------------------------------------

<i>M1</i>	Mass of the ion [amu].
-----------	------------------------

$Z2$	Nuclear charge of the target atom.
$M2$	Mass of the target atom [amu].

Return Value

Returns the maximum energy for Universal screening [keV].

Related Properties and Methods

Calc.ScreeningModel [page 242](#)

Unselect

Procedure Unselect(Z : Integer);

Description

Unselects all cross-sections for all isotopes of element Z .

Parameters

Z	Nuclear charge of the element.
-----	--------------------------------

Return Value

None.

Related Properties and Methods

CrossSec.UnselectAll [page 323](#)

CrossSec.UnselectRutherford [page 324](#)

CrossSec.UnselectSigmaCalc [page 325](#)

UnselectAll

Procedure UnselectAll;

Description

Unselects all cross-sections for all elements in the target.

Parameters

None.

Return Value

None.

Related Properties and Methods

CrossSec.Unselect [page 323](#)

CrossSec.UnselectRutherford [page 324](#)

CrossSec.UnselectSigmaCalc [page 325](#)

UnselectRutherford

Procedure UnselectRutherford(Z : Integer);

Description

Unselects Rutherford cross-sections for all isotopes of element Z. Rutherford cross-sections for scattering (including low energy solution, if kinematically possible) and recoils (if kinematically possible) are unselected.

Parameters

Z	Nuclear charge of the element.
---	--------------------------------

Return Value

None.

Related Properties and Methods

CrossSec.SelectRutherford [page 320](#)

CrossSec.UnselectRutherfordAll [page 324](#)

UnselectRutherfordAll

Procedure UnselectRutherfordAll;

Description

Unselects Rutherford cross-sections for all isotopes of all elements in the target. Rutherford cross-sections for scattering (including low energy solution, if kinematically possible) and recoils (if kinematically possible) are unselected.

Parameters

None.

Return Value

None.

Related Properties and Methods

CrossSec.SelectRutherfordAll [page 321](#)

CrossSec.UnselectRutherford [page 324](#)

UnselectSigmaCalc

Procedure UnselectSigmaCalc(Z : Integer);

Description

Unselects SigmaCalc cross-sections for all isotopes of element Z. The corresponding Rutherford cross-section at lower energies is not unselected.

Parameters

Z	Nuclear charge of the element.
---	--------------------------------

Return Value

None.

Related Properties and Methods

CrossSec.SelectSigmaCalc [page 321](#)

CrossSec.UnselectSigmaCalcAll [page 325](#)

UnselectSigmaCalcAll

Procedure UnselectSigmaCalcAll;

Description

Unselects SigmaCalc cross-sections for all isotopes of all elements in the target. The corresponding Rutherford cross-sections are not unselected.

Parameters

A. OLE automation reference

None.

Return Value

None.

Related Properties and Methods

CrossSec.SelectSigmaCalcAll [page 322](#)

CrossSec.UnselectSigmaCalc [page 325](#)

A.17. Error handling

If SIMNRA is run as stand alone application, i.e. not as OLE server, it reports errors by showing message boxes with error messages or warnings, and program execution is stopped until the *OK* button of the message box is pressed by the user. This behaviour is reasonable for an interactive application, but it is not wishful for an OLE server: The server is controlled by another application or script, and the display of a message box to a script is useless. Therefore, if SIMNRA is running as server, it handles errors in a different way: Message boxes are suppressed and program execution continues, even if an error is encountered. The error is reported by an error flag as return value of the routine which produced it. The text of the last error message can be retrieved with [App.LastMessage](#).

This behaviour can be changed by setting [App.ShowMessages](#) = *true*: In this case error messages will be shown as message boxes, and program execution is stopped until the *OK* button of the box is pressed. SIMNRA is able to detect if it is running as stand alone application or as server, and [App.ShowMessages](#) is set to *false* automatically if invoked as server.

The following code in Visual Basic Script shows the use of the error handling routines:

```
' Create the application object
Set App = CreateObject("Simnra.App")

' Open a XNRA-file
Success = App.Open("c:\temp\test.xnra", 0)

' Some error reported by App.Open: Display last error message and exit
If Not Success Then
    WScript.Echo App.LastMessage
    Exit
End If

' Calculate a spectrum
Success = App.CalculateSpectrum
```

But note that some types of errors will not create error messages. This includes especially:

1. *Incorrect input parameters*: Input parameters, such as [projectile energy](#), [scattering angle](#) etc. are not checked for correctness when supplied through OLE methods, and it is possible to supply physically meaningless parameters such as negative incident energies or scattering angles larger than 180°. This will usually result in hardly predictable errors at a later stage, for example when trying to simulate a spectrum. It is the responsibility of the user to provide physically meaningful input parameters.
2. *Array indices out of range*: Indices of arrayed properties, such as layer numbers in [Target.LayerThickness](#) and other layer properties, can be supplied out of range. Supplied indices were not checked for validity until SIMNRA 7.02 and indices out of range usually resulted in protection faults. This has been changed in SIMNRA 7.03: Supplied indices are now always checked for validity. Writing to arrayed properties with indices out of range is silently ignored. Reading arrayed properties with indices out of range usually returns zero. In both cases there is no error indication.

A.18. Programming examples

A.18.1. Using the OLE automation objects

Sample program in Borland Delphi showing the use of the OLE automation objects.

```
var
  App : Variant;
  Result : Boolean;

begin
  { Create the application object }
  App := CreateOLEObject('Simnra.App');

  { Open a XNRA-file }
  Result := App.Open('c:\temp\test.xnra', 0);

  { Calculate a spectrum }
  Result := App.CalculateSpectrum;

  { Save the XNRA-file }
  Result := App.SaveAs('c:\temp\test.xnra');
end;
```

Sample program in Visual Basic Script showing the use of the OLE automation objects.

A. OLE automation reference

```
' Create the application object
Set App = CreateObject("Simnra.App")

' Open a XNRA-file
Result = App.Open("c:\temp\test.xnra", 0)

' Calculate a spectrum
Result = App.CalculateSpectrum

' Save the NRA-file
Result = App.SaveAs("c:\temp\test.xnra")
```

A.18.2. Using the Fit object

Sample program in Visual Basic Script showing the use of the Fit object.

```
' Create the application object
Set App = CreateObject("Simnra.App")

' Create the fit object
Set Fit = CreateObject("Simnra.Fit")

' Open a XNRA-file
Result = App.Open("c:\temp\test.xnra", 0)

' Fit thickness and composition of layer number 2
Fit.LayerNr = 2
Fit.LayerThickness = True
Fit.LayerComposition = True

' One fit region from channel 100 to 200
Fit.NumberOfRegions = 1
Fit.RegionMinChannel(1) = 100
Fit.RegionMaxChannel(1) = 200

' Perform the fit
Result = App.FitSpectrum
```

A.18.3. Adding layers and elements

Sample program in Visual Basic Script showing how to add layers and elements.

```
' Create the application object
Set App = CreateObject("Simnra.App")
```

```

' Create the target object
Set Target = CreateObject("Simnra.Target")

' Open a XNRA-file
Result = App.Open("c:\temp\test.nra", 0)

' Add an empty layer: Will be the last layer
Target.AddLayer
lay = Target.NumberOfLayers

' Set the layer thickness to 1000E15 atoms/cm2
Target.LayerThickness(lay) = 1000

' Add one element: Will be the last element
Target.AddElement(lay)
el = Target.NumberOfElements(lay)

' Set the element properties
' Attention: Au already has to be present in an already existing layer!
Target.ElementName(lay, el) = "Au"
Target.ElementConcentration(lay, el) = 1.0

' Calculate the spectrum
Result = App.CalculateSpectrum

```

Sample program 1 in Delphi Pascal showing how to add elements using [Target.AddElement](#), [Target.ElementName](#) and [Target.ElementConcentration](#).

```

var
  Target : Variant;
  lay, el : Integer;

begin
  { Create the target object }
  Target := CreateOleObject('Simnra.Target');

  { Add an empty layer }
  Target.AddLayer;

  { Retrieve the current layer number }
  lay := Target.NumberOfLayers;

  { Add one empty element to layer number lay }

```

A. OLE automation reference

```
Target.AddElement(layer);

{ Retrieve the current element number }
el := Target.NumberOfElements[layer];

{ Set the properties of this element }
Target.ElementName[layer, el] := 'Si';
Target.ElementConcentration[layer, el] := 0.5;

{ Add another empty element to layer number layer }
Target.AddElement(layer);

{ Retrieve the current element number }
el := Target.NumberOfElements[layer];

{ Set the properties of this element }
Target.ElementName[layer, el] := 'C';
Target.ElementConcentration[layer, el] := 0.5;
end;
```

Sample program 2 in Delphi Pascal showing how to add elements using [Target.AddElements](#), [Target.ElementZArray](#) and [Target.ElementConcentrationArray](#). Using the array functions is generally faster than adding elements individually, especially for larger numbers of elements.

```
var
  Target : Variant;
  Z : Variant;
  c : Variant;
  layer : Integer;

begin
  { Create the target object }
  Target := CreateOleObject('Simnra.Target');

  { Add an empty layer }
  Target.AddLayer;

  { Retrieve the current layer number }
  layer := Target.NumberOfLayers;

  { Add two empty elements to layer number layer }
  Target.AddElements(layer, 2);
```

```

{ Create a safe variant array of type Integer with 2 elements }
Z := VarArrayCreate([1, 2], varInteger);

{ The first element is silicon, the second element is carbon }
VarArrayPut(Z, 14, [1]);
VarArrayPut(Z, 6, [2]);

{ Set the nuclear charges of both elements in layer number lay }
Target.ElementZArray[lay] := Z;

{ Create a variant array of type Double with 2 elements }
c := VarArrayCreate([1, 2], varDouble);

{ Both elements have atomic concentrations of 0.5 }
VarArrayPut(c, 0.5, [1]);
VarArrayPut(c, 0.5, [2]);

{ Set the concentrations of both elements in layer number lay }
Target.ElementConcentrationArray[lay] := c;
end;

```

Sample program 3 in Delphi Pascal showing how to add elements using [Target.AddElementProperties](#). This function is generally the fastest possibility to add elements, especially for larger numbers of elements.

```

var
  Target : Variant;
  Z : Variant;
  c : Variant;
  lay : Integer;

begin
  { Create the target object }
  Target := CreateOleObject('Simnra.Target');

  { Add an empty layer }
  Target.AddLayer;

  { Retrieve the current layer number }
  lay := Target.NumberOfLayers;

  { Create a safe variant array of type Integer with 2 elements }
  Z := VarArrayCreate([1, 2], varInteger);

```


A. OLE automation reference

```
{ The first element is silicon, the second element is carbon }
VarArrayPut(Z, 14, [1]);
VarArrayPut(Z, 6, [2]);

{ Create a variant array of type Double with 2 elements }
c := VarArrayCreate([1, 2], varDouble);

{ Both elements have atomic concentrations of 0.5 }
VarArrayPut(c, 0.5, [1]);
VarArrayPut(c, 0.5, [2]);

{ Add both elements to layer number lay and set their proprties }
Target.AddElementProperties(lay, Z, c);
end;
```

A.18.4. Using the *Spectrum.GetDataArray* object

Sample program in Delphi Pascal showing how to use the *Spectrum.GetDataArray* object.

```
var
  Spectrum : Variant;
  VarArray : Variant;
  LBound, HBound : Integer;
  Value : Double;
  chan : Integer;

begin
  { Create the spectrum object }
  Spectrum := CreateOLEObject('Simnra.Spectrum');

  { The experimental spectrum is returned as Variant array }
  VarArray := Spectrum.GetDataArray(1);

  { Get low and high array boundaries }
  LBound := VarArrayLowBound(VarArray, 1);
  HBound := VarArrayHighBound(VarArray, 1);

  { Access the data in the Variant array }
  for chan := LBound to HBound do Value := VarArrayGet(VarArray, [chan]);
```

B. SIMNRA windows shell extensions

The windows shell can be extended by shell extensions providing improved integration into the Windows system and additional functionality for custom file extensions. SIMNRA is installed together with two shell extensions providing additional functionality for SIMNRA data files:

- A *thumbnail provider* provides graphical thumbnail images of file contents of `xnra` and `nra` files. These thumbnails are displayed by Windows Explorer.
- A *property handler* provides meta data information for `xnra` and `nra` files. The meta data information can be displayed by Windows Explorer as infotip, in the details pane, in the details view, or in the file details. Windows Search allows searching for specific data files using file contents.

The shell extensions may require a minimum Windows version, see the detailed descriptions for system requirements. Shell extensions are not installed if the minimum system requirement is not fulfilled.

B.1. Thumbnail Provider

SIMNRA is installed together with the thumbnail provider `SIMNRAThumbnailProvider.dll` for `xnra` and `nra` files. The thumbnail provider provides graphical thumbnails of file contents for Windows Explorer and Open or Save dialogs, see [Figure B.1](#). This allows a more graphically oriented work flow with data files.

- The thumbnail provider requires Microsoft Windows Vista or higher. It does not work with Windows XP.
- The thumbnail provider exists as 32-bit and 64-bit version. A 32-bit Windows version requires the 32-bit version of the thumbnail provider, a 64-bit Windows version requires the 64-bit version of the thumbnail provider. The 64-bit version of the thumbnail provider will not run with 32-bit Windows. The 32-bit version of the thumbnail provider will run with Windows 64-bit, but will provide only limited functionality.
- The thumbnail provider is registered for the file extensions `.xnra` and `.nra` during the installation process.
- The thumbnail provider registers itself in
 - `HKEY_CLASSES_ROOT\.xnra\ShellEx\{e357fccd-a995-4576-b01f-234630154e96}`
and

B. SIMNRA windows shell extensions

- HKEY_CLASSES_ROOT\.nra\ShellEx\{e357fcd-a995-4576-b01f-234630154e96}.

You have to delete these registry keys, if you don't want the thumbnails.

B.2. Property handler

SIMNRA is installed together with the property handler `SIMNRAPropertyHandler.dll` for `xnra` and `nra` files. The property handler provides meta data information (incident ion, energy, geometry, etc., see [Figure B.2](#) and [Table B.1](#)) for SIMNRA data files. This meta data information is used for displaying info tips when the mouse rests over a file and can be used by Windows Explorer for displaying additional file information in the details pane, details view, or file details view, see [Figure B.2](#). This meta data information can be also used for searching specific data files using Windows Search. In order to search for file properties using Windows Search either the Property identifier or the Search mnemonic (see [Table B.1](#)) have to be entered into the Search field followed by `:`, i.e. Energy`:`.

When searching for file properties the content of the Search field turns blue, see the Search field in [Figure B.2](#).

Searching for one-letter elements (such as B, C, O, ...) may not give the results you want: If you search for `TargetElements: 0` the search will find not only all data files containing oxygen (O) in the target, but also data files containing Mo or Os. This is a limitation of Windows Search. You have to search for `TargetElements: " 0 "` in order to find only oxygen.

- The property handler requires Microsoft Windows Vista or higher. It does not work with Windows XP.
- Search mnemonics are only available in Windows 7 and higher. In Windows Vista the Property identifiers have to be used.
- The property handler exists as 32-bit and 64-bit version. A 32-bit Windows version requires the 32-bit version of the property handler, a 64-bit Windows version requires the 64-bit version of the property handler. The 64-bit version of the property handler will not run with 32-bit Windows. The 32-bit version of the property handler will run with Windows 64-bit, but will provide only limited functionality.
- The property handler is registered for the file extensions `.xnra` and `.nra` during the installation process.

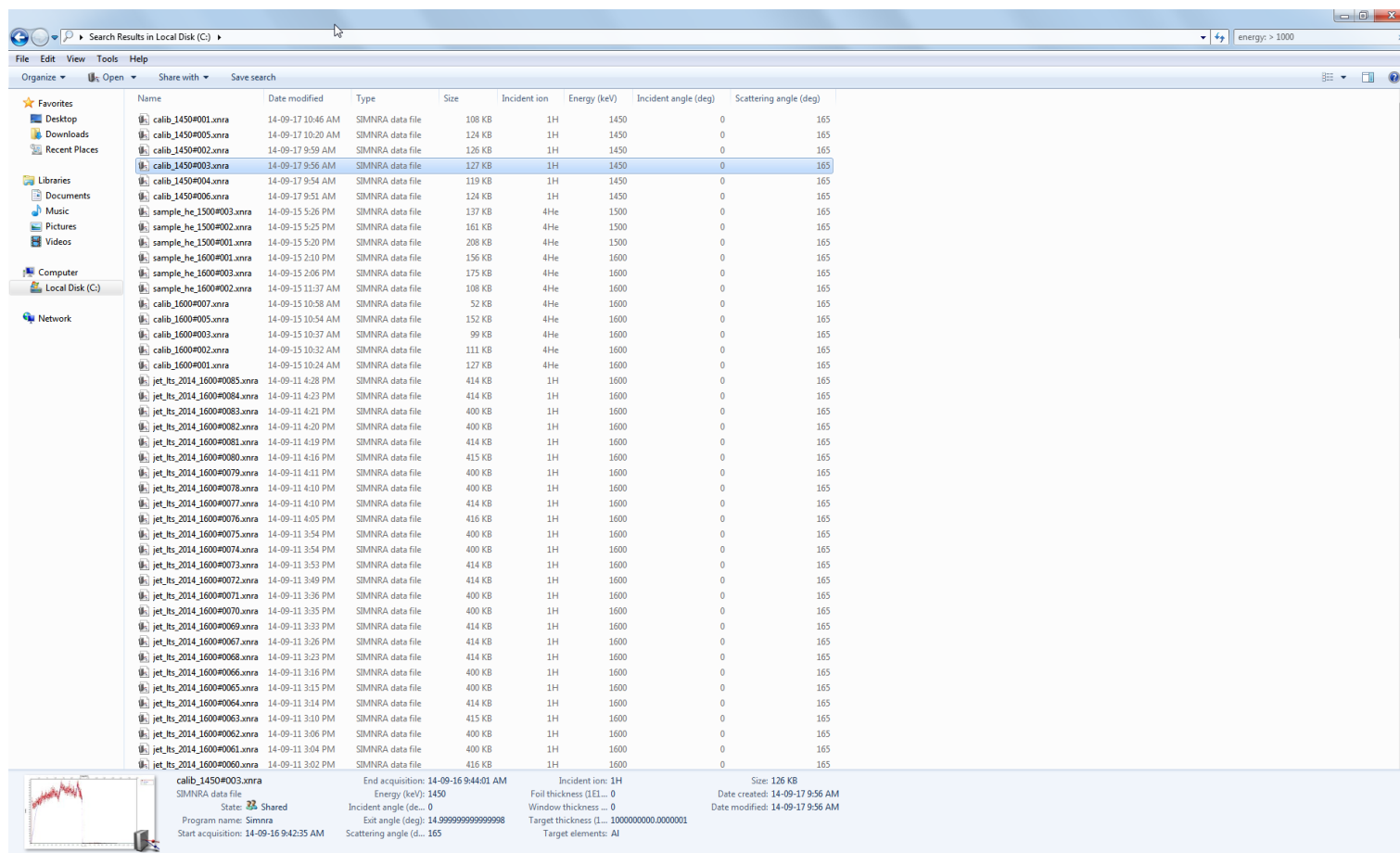


Figure B.2.: Windows Explorer view with file details of SIMNRA data files. Only data files with incident energy larger than 1000 keV are displayed, see the Search field in the upper right corner. Additional information about the file content and a thumbnail image are displayed in the details pane at the bottom.

Table B.1.: Available properties of SIMNRA data files through the SIMNRA property handler. The explorer label is displayed in Windows Explorer, the search mnemonic can be used in Windows Search (see Figure B.2). The property identifier should be used when accessing the property with third party programs and can be used in Windows Search instead of the search mnemonic.

Property identifier	Explorer label	Search mnemonic	Description
IPP.Simnra.Ion	Incident ion	IncidentIon	Mass and name of incident ion, for example 4He.
IPP.Simnra.Energy	Energy (keV)	Energy	Energy of incident ion, in keV.
IPP.Simnra.IncidentAngle	Incident angle (deg)	IncidentAngle	Incident angle, in deg.
IPP.Simnra.ExitAngle	Exit angle (deg)	ExitAngle	Exit angle, in deg.
IPP.Simnra.ScatteringAngle	Scattering angle (deg)	ScatteringAngle	Scattering angle, in deg.
IPP.Simnra.TargetThickness	Target thickness (1E15 at/cm2)	TargetThickness	Total thickness of the target, in 10^{15} atoms/cm ² .
IPP.Simnra.TargetElements	Target elements	TargetElements	List of elements in the target. The list is ordered with decreasing total amount of the elements.
IPP.Simnra.FoilThickness	Foil thickness (1E15 at/cm2)	FoilThickness	Total thickness of the absorber foil, in 10^{15} atoms/cm ² .
IPP.Simnra.FoilElements	Foil elements	FoilElements	List of elements in the absorber foil. The list is ordered with decreasing total amount of the elements.
IPP.Simnra.WindowThickness	Window thickness (1E15 at/cm2)	WindowThickness	Total thickness of the beam window, in 10^{15} atoms/cm ² .
IPP.Simnra.WindowElements	Window elements	WindowElements	List of elements in the beam window. The list is ordered with decreasing total amount of the elements.
IPP.Simnra.StartAcquisition	Start acquisition	StartAcquisition	Starting date and time of the acquisition of experimental data contained in the file.
IPP.Simnra.EndAcquisition	End acquisition	EndAcquisition	Ending date and time of the acquisition of experimental data contained in the file.

C. SIMNRA access keys and keyboard shortcuts

A combination of keystrokes can be used to quickly perform an action without having to navigate to the menu. The key combination consists of a modifier key (such as CTRL or ALT) and an accompanying key (such as a character key or a function key).

- **Access keys:** Access keys allow to choose menu items and perform menu actions without having to navigate to the menu first. Access keys are indicated by underlining one of the characters in the menu. The menu item is then activated by pressing the ALT key along with the underlined character. Access keys are not case sensitive. For example, pressing ALT → F → O activates the *File:Open...* command.
- **Keyboard shortcuts:** Keyboard shortcuts invoke a command directly without having to navigate to the menu. Some commands are only available as keyboard shortcuts. All SIMNRA keyboard shortcuts are listed in [Table C.1](#).

Table C.1.: List of SIMNRA keyboard shortcuts.

Hot Key	Menu/Action	Context
Editing		
CTRL + A	–	Marks the whole content of an Edit field.
CTRL + C	–	Copies the marked content of an Edit field to the clipboard.
CTRL + V	–	Pastes the content of the clipboard to an Edit field.
Main form		
CTRL + D	<i>Tools: Data reader...</i>	Opens the data reader tool.
CTRL + I	<i>Tools: Integrate spectrum...</i>	Opens the integrate spectrum tool.
CTRL + J	<i>Edit: Copy page</i>	Copies experimental and simulated spectra to the clipboard.
CTRL + K	<i>Edit: Copy data</i>	Copies experimental and simulated spectra to the clipboard.
CTRL + N	<i>File: New</i>	Resets the program to its starting values.
CTRL + O	<i>File: Open...</i>	Opens the <i>File:Open...</i> dialog.
CTRL + P	<i>File: Print...</i>	Opens the <i>File:Print...</i> dialog.
CTRL + Q	<i>File: Exit</i>	Exits the program.
CTRL + S	<i>File: Save</i>	Opens the <i>File:Save</i> dialog.
CTRL + X	<i>Plot: x-axis...</i>	Properties of the x-axis.
CTRL + Y	<i>Plot: y-axis...</i>	Properties of the y-axis.
CTRL + ALT + X	<i>Plot: Rescale x-axis...</i>	Rescales the x-axis.
CTRL + ALT + Y	<i>Plot: Rescale y-axis...</i>	Rescales the y-axis.
F1	<i>Help: User's guide</i>	Opens the SIMNRA User's Guide.
F9	<i>Calculate: Calculate spectrum</i>	Calculates the simulated spectrum.
F10	<i>Calculate: Calculate spectrum fast</i>	Calculates the simulated spectrum with decreased accuracy, but faster.
SHIFT + F10	–	Activates the context menu of the plot, see section 3.10 .
ALT	–	Changes to the main menu and shows access keys, see section C .
ESC	<i>Calculate: Abort calculation</i>	Aborts a running calculation.
ALT + F4	<i>File: Exit</i>	Exits the program.
Target form		
CTRL + L	<i>Edit: Copy layer</i>	Copies a layer to the clipboard.
CTRL + ALT + L	<i>Edit: Paste layer</i>	Pastes a layer from the clipboard.

D. The R33 cross section file format

For recent updates to the r33 file format see www-nds.iaea.org/exfor/x4guide/R33/.

The R33 cross section file format

April 2002

By I. C. Vickridge

Groupe de Physique des Solides, UMR 7588 du CNRS

Tour 23, Universités de Paris 7 et 6

2, Place Jussieu

75251 Paris

D.1. Introduction

In September 1991, in response to the workshop on cross sections for Ion Beam Analysis (IBA) held in Namur (July 1991, Nuclear Instruments and Methods B66(1992)), a simple ascii format was proposed to facilitate transfer and collation of nuclear reaction cross section data for Ion Beam Analysis (IBA) and especially for Nuclear Reaction Analysis (NRA). Although intended only as a discussion document, the ascii format - referred to as the R33 (DSIR Report 33) format - has become a de facto standard. In the decade since this first proposal there have been spectacular advances in computing power and in software usability, however the simplicity and cross-platform compatibility of the ascii character set has ensured that the need for an ascii format remains.

Nuclear reaction cross section data for Nuclear Reaction analysis has been collected and archived on the Sigmabase websites (google : Sigmabase) for about the last 7 years. This data has largely been entered in the R33 format, although there is a series of elastic cross sections that are expressed as the ratio to the corresponding Rutherford cross sections that have been entered in a format referred to as RTR (ratio to Rutherford). During this time the R33 format has been modified and added to - firstly to take into account angular distributions, which were not catered for in the first proposal, and more recently to cater for elastic cross sections expressed as the ratio-to-Rutherford, which it is useful to have for some elastic scattering programs. It is thus timely to formally update the R33 format.

There exists also the large nuclear cross section data collections of the Nuclear Data Network - the OECD NEA Nuclear data section, the IAEA Nuclear data section, and the Brookhaven National Laboratory National Nuclear Data Centre amongst others. The R33 format is proposed

D. The R33 cross section file format

to become a legal computational format for the Nuclear Data Network ('Nuclear Data Needs in Ion Beam Analysis'. I.C. Vickridge. In "Long Term Needs for Nuclear Data Development", Report INDC(NDS)-428, August 2001, International Atomic Energy Agency, Vienna.). It is thus also necessary to provide an updated formal definition of the R33 format in order to provide the necessary specification for adoption of R33 as an accepted computational format.

Guiding considerations. In defining the updated R33 format I have required that previous valid R33 files should also conform to the updated format. This is so that R33 reading programs that conform to the updated specification will be able to read the existing R33 files - providing backward compatibility. There is also some redundancy in the format. This is partly from intellectual laziness, but also provides some checking of internal consistency to weed out errors.

D.2. The new R33 Format definition.

An R33 data file contains one cross section either as a function of (laboratory) incident energy, or as a function of (laboratory) detection angle. The file is made up of entries, and the data section. Each entry consists of a legal keyword followed by a colon followed by a space, followed by data in Ascii format. The keyword may be in any mixture of upper and lower case characters. Legal separators for numerical data are space, comma, colon, and semi-colon characters. Decimal points are represented only by full stops, and not by commas (as can be the case in some European countries). The legal ascii character set for the purposes of R33 files is ascii 0 to ascii FF. Apart from the 'Comment' entry and the optional 'Version' entry, entries may be in any order. Each entry ends with a carriage-return line-feed sequence. Some entries are optional [O], most are required [R], and some are mutually exclusive [Mx] where all entries for which the value of x is the same are mutually exclusive. See below for special conditions that apply to the keywords 'Comment', 'Nvalues', and 'Data', and 'EndData'. Default values are suggested for R33 reading routines, so that if an optional entry is omitted, or if a required entry is unreadable or missing illegally, the value of the corresponding variable in the reader is well-defined. All energies are expressed for the laboratory frame in keV and all angles in the laboratory frame in degrees. English is the preferred language.

Special note on data entry order: The original R33 specification called for data to be listed as x, y, xerror, yerror, however the R33 files originally generated for the Sigbase contained data entries in the order : x, xerror, y, yerror. The original specification was intended to allow for files containing no error information to be smaller, since the two final entries could simply be omitted. However given that the Sigbase data fits easily on a single floppy, without compression, the file size arguments are not compelling and insisting on following the original specification will involve disrupting several existing readers, as well as introducing confusion through the existence of two families of R33 files since copies of the erroneous files will probably lie around for years in different places. So, the new specification legalises the previous erroneous usage.

D.2.1. Syntax of an R33 Entry

The syntax for the list of legal keywords and the associated data is :

Keyword: [Mx, O/R] (Default) <data type>
Note: Additional notes and guidelines concerning use of the entry. Data type may be:

string (an arbitrary series of ascii characters)

n (integer) a series of ascii characters without decimal point representing a signed integer number

r (real) a series of ascii characters that represent a signed real number. Format is fairly flexible - but only decimal points (and NOT decimal commas) are accepted. Any format that can be read in a Borland Pascal readln(r) statement is acceptable.

D.2.2. List of legal entries

Comment: [R] ('None') <String>
Note: An unlimited number of ascii characters, including single CR LF sequences, but terminating by a double CR LF sequence. There is no requirement to embed CR LF sequences within the Comment, however it is recommended to place CR LF sequences at convenient places at least every 80 characters so that if the file is printed, the comment field is printed on successive lines that are not longer than 80 characters. The double CR LF sequence that signals the end of the comment is simply a blank line. It is not felt necessary to specify an upper limit to the size of the comment, however it is expected that a useful comment would not be longer than a few tens of lines - or a few thousand characters. Note that Unix systems place only a LF character to signal an end-of-line. This is illegal for R33 files. There are freeware and shareware utilities that can add the necessary CR characters if R33 files are generated under Unix.

Version: [O] ('R33') <String>
Note: Allowed values are (case-insensitive) 'R33' and 'R33a'. This entry can be used to signal that the file conforms to the special subset of R33 files proposed by M. O. Thompson for elastic scattering cross sections for use in RUMP. Sigbase files will always be DSIR R33, but the 'R33a' variant is detailed here for completeness. All R33a files are legal R33 files, but R33a files have the following additional conditions:
1) The Version entry is required, and must be the first entry after the Comment.
2) Only elastic cross sections can be in valid R33a files.
3) Nvalues must either not be present (so that Data and EndData entries are used), or have a value of less-than-or-equal-to zero.

Source: [R]('Unknown')<String>

D. The R33 cross section file format

Note: A concise bibliographic source (preferable) or another indication of where the data has come from (avoid if possible). This field should contain the most authoritative original source for the data. This will usually be the original publication, or thesis reference. In some cases data has been input by experimenters before or without publication. In this case this entry should contain something like 'Measured and input by D. Withrington'. It should be kept small - an upper limit of 256 characters is suggested, but not required. It would be expected that further details pertaining to Mrs Withrington would be found in the Comment.

Name: [R]('unknown')<String>
Note: The name of person or institution responsible for creating the R33 file. For R33 files automatically created from Exfor files, this would be IAEA, or NNDC, or International Nuclear Data Network or whatever. No provision is made here for including update histories, however this may be accommodated in the Comment field.

Address1: [O] (' ')<String>

⋮

Address9: [O] (' ') <String>
Note: The address of the person or institution responsible for creating the R33 file. Up to nine lines of address information may be included. This can include telephone numbers, emails and so on.

Serial Number: [R] (0) <n>
Note: The serial number will be a number providing a unique link back to the Exfor dataset from which the R33 file was generated. The default value of zero means that this number has not been assigned.

Reaction: [R] (' ') <String>

D.2. The new R33 Format definition.

Note: the reaction string is written in a standard format that can be parsed without too much difficulty. It conforms to the usual notation of:

target nucleus(incident ion, light product)product nucleus

Nuclei are specified by their chemical symbol preceded by A : e.g. ^{28}Si , or ^6Li . The mass number is required. Some common light species may also be represented by shorthand notation, with lower case being required.

n=neutron

p=proton

d=deuterium

t=triton

a=alpha particle

h= ^3He

x=x-ray

g=gamma

The light product may correspond to a particular energy level of the product nucleus. This is signalled by a 'postfix' on the light product. For example, $^{16}\text{O}(\text{d},\text{p}1)^{17}\text{O}$ is the (d,p) reaction which leaves the residual 17 oxygen in the 1st excited state. Some cross sections may be sums of several particle groups corresponding to different excited states of the compound or product nucleus. Usually such a cross section would be used when the particle groups are not resolved by the detection method employed. In this case, the postfix lists the states concerned, separated by plus signs. E.g. $^{14}\text{N}(\text{d},\text{p}5+6)^{15}\text{N}$. Some elastic cross sections correspond to targets having several isotopes. In this case, it is necessary to use the 'composition' keyword.

Masses:

[R] (1,1,1,1) <r,r,r,r>

Note: Four mass values in amu corresponding to the four nuclei specified in the reaction string, separated by legal separators. The order is m_1 , m_2 , m_3 , m_4 for a reaction in which $m_1+m_2 \rightarrow m_3+m_4$. The specification of which of the two initial and final masses are m_1 and m_3 respectively is given by the reaction string in which we always have : $m_2(m_1, m_3,)m_4$, so that m_1 corresponds to the projectile and m_3 to the light product. At present there is no intention to cater for the few cases in which there are three or more products - for example $^{11}\text{B}(\text{p},\text{a})2\text{a}$. In principle the masses could be deduced directly from the reaction string, however in the interests of simplicity it seems worthwhile adding them to the R33 file to avoid having to write a reaction string parser in R33 readers. In the special case where the 'composition' keyword is used, the values there override any contained in the 'Masses' entry.

Zeds:

[R] (1,1,1,1) <n,n,n,n>

Note: Four integers representing the atomic number of the four nuclei specified in the reaction string, in the same order as the mass entries. The Zs could also be deduced directly from the reaction string, but see the comment in the Masses entry.

Composition: [O] ('Natural') <string, r, string, r,...>

D. The R33 cross section file format

Note: This entry caters for elastic cross sections measured from targets that contain a mixture of isotopes from which the elastically particles are not resolved. It consists of a list of isotopes and atom-proportions, or the value 'natural' (case insensitive) which means that a target of naturally occurring isotopic composition has been used. Isotopes are specified as for the reaction string, without shorthand notation. Example:

Target: 12C, 23.0, 13C, 26.0

If the proportions do not sum to 100, then it is assumed that they are relative amounts. In the example given, 23/49 of the atoms are 12C, and 26/49 are 13C. If the 'Composition' entry exists and has a legal value, then the values in the 'masses' entry should correspond to the appropriate weighted sum indicated in the 'composition' keyword. For example, if the target for an elastic cross section is natural silicon, then the mass given should be 28.086 - the weighted sum of stable Si isotopes in natural abundance.

Qvalue: [R] (0.0) <r, r, r, r, r>
Note: A list of up to five Q values, expressed in keV, separated by legal separators. As explained in the Reaction entry, some cross sections are for multiple particle groups, for example when the groups are not resolved experimentally. In this case a Q value is required for each particle contributing to the cross section.

Distribution: [R] ('Energy') <String>
Note: Allowed values are 'Energy', and 'Angle'. This entry says whether the data contained in this file are for a cross-section as a function of laboratory energy ('Energy') or laboratory angle ('Angle').

Theta: [M1, R] (0.0) <r>
Energy: [M1,R](1.0) <r>
Note: Theta gives the laboratory angle with respect to the incident beam, so that backscattering would be 180°. Energy gives the laboratory energy of the incident beam. If both entries are present (an illegal condition ...) then only the entry corresponding to the keyword in the 'Distribution' entry will be used.

Sigfactors: [O] (1.0, 0.0) <r,r>
Note: Scale conversion factor and its associated error common to all the cross section data. See original R33 publication in Appendix 1 for discussion.

Units: [O] ('mb') <String>

D.2. The new R33 Format definition.

Note: Valid values are 'mb', 'rr' and 'tot'. R33 files are always in mb/sr, or units proportional to mb/sr for differential cross sections, and mb for total cross sections. The constant of proportionality given in the enfactors entry is independent of energy. Some users find it preferable to have elastic scattering cross sections relative to the Rutherford cross section. Since the conversion factor is no longer energy-independent, the enfactors entry can no longer cater for this. In this case, the units entry should specify 'rr' for ratio to Rutherford. Nevertheless, it is recommended that elastic cross sections be stored as cross sections just like the inelastic scattering cross sections - i.e. in mb/sr. The value 'tot' indicates that the cross section is integrated over all angles, and is expressed as a function of energy. Thus if 'tot' is used as a unit then the distribution must be 'energy' and the value of theta has no meaning. Nevertheless, it is suggested that a valid real number be given for theta so that reading routines don't have to cater for non-numerical values for theta (such as 'Theta: irrelevant!').

Enfactors: [O] (1.0, 0.0, 0.0, 0.0) <r,r,r,r>
Note: Scale conversion factors and associated errors common to all the energy or angle data. See original R33 publication in Appendix 1 for discussion.

Nvalues: [M2,R] (0)<n>

Data: [M2,R]

Enddata: [O]

Note: two methods are allowed for representing the data. The first corresponds to the original R33 specification. The data immediately following the 'Nvalues' entry consists of the cross section data, one point per line, and each point represented by four values (X, dX, Y, dY):


energy(or angle), energy(or angle) random error, sigma, sigma random error

The data ends after Nvalues lines of data.

Alternatively (and recommended), the data may be bracketed by 'Data' and 'Enddata' entries. An entry of 'Nvalues: 0' is equivalent to a 'Data' entry. The data immediately follow the Data entry, one point per line as for the Nvalues option, and the file terminates with the end of the file, or with the optional EndData entry. Nvalues is maintained for backward compatibility, but in practice most routines will simply read and count the number of lines read until the end of the file or an EndData entry is reached, so this is the preferred option. The use of the Enddata entry simply allows a check that all of the data values are contained in the file and have been read. Data entries must be arranged in order of increasing energy or angle. Duplicate energy or angle values are not allowed (the cross section must be single-valued).

E. Origin fitting function for the two-piece normal distribution

A fitting function for the two-piece normal distribution (asymmetric Gaussian function) for Microcal Origin is attached. The function has been tested with Origin 6.0 to 8.5.

Click the following icon with the right mouse button and select *Save to disc...*  See [subsection 3.5.1](#) and [Figure 3.7](#) for the definition and a graphical representation of this function. See below for installation and usage instructions.

E.1. Installation in Origin

1. Copy the file `asymGauss.fdf` into the `FitFunc` directory of your Origin installation (i.e. where all other *.fdf files are).
2. Navigate to the Origin base directory. Edit the file `Nlsf.ini` with a text editor, for example Notepad. Navigate to the [Peak Functions] section in this file, and add the line `asymGauss=asymGauss`. Save the modified `Nlsf.ini`.

E.2. Usage in Origin

1. Select "Analysis:Non linear curve fit" (or Ctrl+Y)
2. Press the "More..." button in the "NonLinear Curve Fitting" form.
3. Select "Peak Functions" in "Categories", and "asymGauss" in "Functions".
4. Start fitting. The parameters have the following meaning:
 - `xc` Mode (x-position of the maximum) of the distribution.
 - `A` Area below the distribution.
 - `h1` Half width at half maximum (HWHM) towards low numbers.
 - `h2` HWHM towards high numbers.

Bibliography

- [1] R. Amirikas, D.N. Jamieson, and S.P. Dooley. “Measurement of the (p,p) elastic cross sections for C, O and Si in the energy range 1.0–3.5 MeV”. In: *Nucl. Instr. Meth. B* 77 (1993), p. 110 (cit. on pp. [139](#), [173](#)).
- [2] G. Amsel, G. Battistig, and A. L’Hoir. “Small Angle Multiple Scattering of fast Ions, Physics, Stochastic Theory and Numerical Calculations”. In: *Nucl. Instr. Meth. B* 201 (2003), p. 325. DOI: [10.1016/S0168-583X\(02\)01740-8](#) (cit. on pp. [32](#), [116](#), [117](#), [120](#), [123](#), [126](#)).
- [3] G. Amsel, E. d’Artemare, G. Battistig, V. Morazzani, and C. Ortega. “A characterisation of the morphology of porous silicon films by proton energy loss fluctuation measurements with a narrow resonance in the $^{15}\text{N}(\text{p},\alpha\gamma)^{12}\text{C}$ reaction”. In: *Nucl. Instr. Meth. B* 122 (1997), p. 99 (cit. on p. [148](#)).
- [4] G. Amsel, E. Girard, G. Vizkelethy, G. Battistig, Y. Girard, et al. “High pulse rate and pileup handling in precision RBS”. In: *Nucl. Instr. Meth. B* 64 (1992), p. 811 (cit. on p. [156](#)).
- [5] H.H. Andersen, F. Besenbacher, P. Loftager, and W. Möller. “Large-angle scattering of light ions in the weakly screened Rutherford region”. In: *Phys. Rev. A* 21.6 (1980), p. 1891 (cit. on pp. [84](#), [85](#), [168](#)).
- [6] H.H. Andersen and J.F. Ziegler. *Hydrogen - Stopping Powers and Ranges in All Elements*. Vol. 3. The Stopping and Ranges of Ions in Matter. New York: Pergamon Press, 1977 (cit. on pp. [31](#), [34](#), [96](#), [100](#), [123](#), [168](#)).
- [7] E.G. Androulakaki, M. Kokkoris, M. Mayer, E. Mitsi, N. Patronis, et al. “A comparative study of multiple scattering calculations implemented in general purpose Monte Carlo and selected ion beam analysis codes”. In: *Nucl. Instr. Meth. B* 496 (2021), p. 71 (cit. on p. [169](#)).
- [8] B.H. Armitage, J.D.F. Ramsay, and F.P. Brady. “The measurement of pore size in porous and microporous materials using resonant ion beam backscattering”. In: *Nucl. Instr. Meth.* 149 (1978), p. 329 (cit. on p. [148](#)).
- [9] K. Arstila, T. Sajavaara, and J. Keinonen. “Monte Carlo Simulation of Multiple and Plural Scattering in Elastic Recoil Detection”. In: *Nucl. Instr. Meth. B* 174 (2001), p. 163 (cit. on p. [120](#)).
- [10] G. Audi and A.H. Wapstra. “The 1995 Update to the Atomic Mass Evaluation”. In: *Nucl. Phys. A* 595 (1995), p. 409 (cit. on p. [81](#)).
- [11] A. Azzalini. “A Class of Distributions Which Includes the Normal Ones”. In: *Scandinavian Journal of Statistics* 12 (1985), p. 171 (cit. on p. [120](#)).

- [12] J.E.E. Baglin, A.J. Kellog, M.A. Crockett, and A.H. Shih. “Absolute cross section for hydrogen forward scattering”. In: *Nucl. Instr. Meth. B* 64 (1992), p. 469 (cit. on p. 175).
- [13] N. Pessoa Barradas, J. Cruz, M. Fonseca, A.P. de Jesus, A. Lagoyannis, et al. “International Atomic Energy Agency inter-comparison of particle induced gamma-ray emission codes for bulk samples”. In: *Nucl. Instr. Meth. B* 468 (2020), p. 37. doi: [10.1016/j.nimb.2020.02.019](https://doi.org/10.1016/j.nimb.2020.02.019) (cit. on pp. iv, 61).
- [14] N.P. Barradas, E. Alves, and D. Babonneau. “Characterization of FePt nanoparticles in FePt/C multilayers”. In: *Nucl. Instr. Meth. B* 219-220 (2004), p. 919 (cit. on p. 148).
- [15] N.P. Barradas, E. Alves, C. Jeynes, and M. Tosaki. “Accurate simulation of backscattering spectra in the presence of sharp resonances”. In: *Nucl. Instr. Meth. B* 247 (2006), p. 381 (cit. on p. 91).
- [16] N.P. Barradas, K. Arstila, G. Battistig, M. Bianconi, N. Dytlewski, et al. “International Atomic Energy Agency intercomparison of ion beam analysis software”. In: *Nucl. Instr. Meth. B* 262 (2007), p. 281. doi: [10.1016/j.nimb.2007.05.018](https://doi.org/10.1016/j.nimb.2007.05.018). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0027-0732-B>. URL: [https://mam.home.ipp.mpg.de/Barradas-Nuclear-Instruments-Methods-B-262-\(2007\)-281.pdf](https://mam.home.ipp.mpg.de/Barradas-Nuclear-Instruments-Methods-B-262-(2007)-281.pdf) (cit. on p. iii).
- [17] N.P. Barradas, M. Mayer, and M. Thompson. “A new ion beam analysis data format”. In: *Nucl. Instr. Meth. B* 268 (2010), p. 1824. doi: [10.1016/j.nimb.2010.02.093](https://doi.org/10.1016/j.nimb.2010.02.093). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0026-EED0-D>. URL: [https://mam.home.ipp.mpg.de/Barradas-Nuclear-Instruments-Methods-B-268-\(2010\)-1824.pdf](https://mam.home.ipp.mpg.de/Barradas-Nuclear-Instruments-Methods-B-268-(2010)-1824.pdf) (cit. on pp. iv, 11, 12, 65, 76).
- [18] N.P. Barradas, R.P. Pezzi, and I.J.R. Baumvol. “Use of the gamma function for straggling in simulation of RBS spectra”. In: *Nucl. Instr. Meth. B* 261 (2007), p. 422 (cit. on p. 123).
- [19] P. Bauer, E. Steinbauer, and J.P. Biersack. “Rutherford backscattering beyond the single scattering model”. In: *Nucl. Instr. Meth. B* 79 (1993), p. 443. doi: [10.1016/0168-583X\(93\)95383-G](https://doi.org/10.1016/0168-583X(93)95383-G) (cit. on pp. 116, 170).
- [20] R. Behrisch, S. Grigull, U. Kreissig, and R. Grötschel. “Influence of Surface Roughness on Measuring Depth Profiles and the Total Amount of Implanted Ions by RBS and ERDA”. In: *Nucl. Instr. Meth. B* 136-138 (1998), p. 628. doi: [10.1016/S0168-583X\(97\)00798-2](https://doi.org/10.1016/S0168-583X(97)00798-2) (cit. on p. 130).
- [21] F. Besenbacher, I. Stensgaard, and P. Vase. “Absolute Cross Section for Recoil Detection of Deuterium”. In: *Nucl. Instr. Meth. B* 15 (1986), p. 459 (cit. on p. 175).
- [22] P. De Bievre and P.D.P. Tylor. In: *Int. J. Mass. Spectrom. Ion Phys.* 123 (1993), p. 149 (cit. on p. 81).
- [23] J.R. Bird, P. Duerden, D.D. Cohen, G.B. Smith, and P. Hillery. “Depth Profiles and Microtopology”. In: *Nucl. Instr. Meth.* 218 (1983), p. 53 (cit. on p. 130).
- [24] J.R. Bird and J.S. Williams, eds. *Ion Beams for Materials Analysis*. Sydney, New York, Tokyo: Academic Press, 1989 (cit. on p. 103).
- [25] N. Bohr. “The Penetration of Atomic Particles through Matter”. In: *Mat. Fys. Medd. Dan. Vid. Selsk.* 18.8 (1948) (cit. on pp. 103, 104, 122, 168).

- [26] D. Boutard, W. Möller, and B.M.U. Scherzer. “Influence of H-C bonds on the stopping power of hard and soft carbonized layers”. In: *Phys. Rev. B* 38.5 (1988), p. 2988 (cit. on p. 100).
- [27] M. Bozoian. “Deviations from Rutherford backscattering for $Z = 1, 2$ projectiles”. In: *Nucl. Instr. Meth. B* 58 (1991), p. 127 (cit. on p. 87).
- [28] M. Bozoian. “A useful formula for departures from Rutherford backscattering”. In: *Nucl. Instr. Meth. B* 82 (1993), p. 602 (cit. on p. 87).
- [29] M. Bozoian, K.M. Hubbard, and M. Nastasi. “Deviations from Rutherford-scattering cross sections”. In: *Nucl. Instr. Meth. B* 51 (1990), p. 311 (cit. on p. 87).
- [30] W.H. Bragg and R. Kleeman. In: *Philos. Mag.* 10 (1905), p. 318 (cit. on p. 100).
- [31] D.A. Bromley, J.A. Kuehner, and E. Almqvist. “Resonant Elastic Scattering of C^{12} by Carbon”. In: *Phys. Rev. Lett.* 4.7 (1960), p. 365 (cit. on p. 88).
- [32] D.A. Bromley, J.A. Kuehner, and E. Almqvist. “Elastic Scattering of Identical Spin-Zero Nuclei”. In: *Phys. Rev.* 123.3 (1961), p. 878 (cit. on pp. 88, 89).
- [33] M.S. Caceci and W.P. Cacheris. “Fitting Curves to Data”. In: *Byte* 5 (1984), p. 340 (cit. on p. 49).
- [34] D. Cano-Ott, J.L. Tain, A. Gadea, B. Rubio, L. Batist, et al. “Pulse pileup correction of large NaI(Tl) total absorption spectra using the true pulse shape”. In: *Nucl. Instr. Meth. A* 430 (1999), p. 488 (cit. on p. 158).
- [35] W.K. Chu. In: *Phys. Rev.* 13 (1976), p. 2057 (cit. on pp. 104, 122, 168).
- [36] L. Cliche, S.C. Gujrathi, and L.A. Hamel. “Pulse Height Defects for ^{16}O , ^{35}Cl and ^{81}Br Ions in Silicon Surface Barrier Detectors”. In: *Nucl. Instr. Meth. B* 45 (1990), p. 270 (cit. on p. 74).
- [37] J.P. Coad, B.G. Skorodumov, V.G. Ulanov, and C.H. Wu. “Hydrogen isotope analysis of thick layers deposited in tokamaks”. In: *Vacuum* 47 (1996), p. 985 (cit. on p. 62).
- [38] D. Dieumegard, D. Dubreuil, and G. Amsel. In: *Nucl. Instr. Meth.* 166 (1979), p. 431 (cit. on p. 113).
- [39] D.R. Dixon, G.L. Jensen, S.M. Morrill, R.L. Walter, and C.R. Gould. “Multiple scattering of protons and deuterons by thick foils”. In: *Nucl. Instr. Meth.* 213 (1983), p. 524 (cit. on p. 117).
- [40] L.R. Doolittle. “Algorithms for the rapid simulation of Rutherford backscattering spectra”. In: *Nucl. Instr. Meth. B* 9 (1985), p. 344. DOI: [10.1016/0168-583X\(85\)90762-1](https://doi.org/10.1016/0168-583X(85)90762-1) (cit. on pp. 79, 93, 130, 152).
- [41] L.R. Doolittle. “A semiautomatic algorithm for Rutherford backscattering analysis”. In: *Nucl. Instr. Meth. B* 15 (1986), p. 227. DOI: [10.1016/0168-583X\(86\)90291-0](https://doi.org/10.1016/0168-583X(86)90291-0) (cit. on pp. 79, 130).
- [42] W. Eckstein and M. Mayer. “Rutherford backscattering from layered structures beyond the single scattering model”. In: *Nucl. Instr. Meth. B* 153 (1999), p. 337. DOI: [10.1016/S0168-583X\(98\)01011-8](https://doi.org/10.1016/S0168-583X(98)01011-8) (cit. on pp. iii, 116, 169, 170).

- [43] R.D. Edge and U. Bill. “Surface Topology using Rutherford Backscattering”. In: *Nucl. Instr. Meth.* 168 (1980), p. 157 (cit. on p. 130).
- [44] EG&G ORTEC. *Model 672 Spectroscopy Amplifier Operating Manual*. Tech. rep. Oak Ridge, U.S.A.: EG&G ORTEC (cit. on p. 27).
- [45] A. Fonseca, N. Franco, E. Alves, N.P. Barradas, J.P. Leita, et al. “High resolution backscattering studies of nanostructured magnetic and semiconducting materials”. In: *Nucl. Instr. Meth. B* 241 (2005), p. 454 (cit. on p. 148).
- [46] B.R. Frieden. *Probability, Statistical Optics, and Data Testing: A Problem Solving Approach*. Ed. by T.S. Huang. Vol. 10. Springer Series in Information Sciences. Berlin: Springer, 1991. ISBN: 3-540-53310-9 (cit. on p. 120).
- [47] J.S. Garvin and S.I. McClean. “Convolution and sampling theory of the binormal distribution as a prerequisite to its application in statistical process control”. In: *The Statistician* 46 (1997), p. 33 (cit. on p. 121).
- [48] Group of Nuclear Physics. *MottCalc*. <http://nuclearphysics.ntua.gr/downloads.php> (cit. on p. 60).
- [49] A. Gurbich. “SigmaCalc recent development and present status of the evaluated cross-sections for IBA”. In: *Nucl. Instr. Meth. B* 371 (2016), p. 27 (cit. on p. 169).
- [50] A.F. Gurbich. “Evaluation of non-Rutherford proton elastic scattering cross section for carbon”. In: *Nucl. Instr. Meth. B* 136-138 (1998), p. 60. DOI: [10.1016/S0168-583X\(97\)00837-9](https://doi.org/10.1016/S0168-583X(97)00837-9) (cit. on pp. 91, 92).
- [51] A.F. Gurbich. “On the concept of an actual Coulomb barrier”. In: *Nucl. Instr. Meth. B* 217 (2004), p. 183 (cit. on p. 169).
- [52] M. Hautala and M. Luomajärvi. “Corrections to the Rutherford scattering cross section in the backscattering analysis”. In: *Rad. Effects* 45 (1980), p. 159 (cit. on p. 84).
- [53] Z.J. He, L.Q. Shi, C.Z. Liu, L. Zhang, Y.F. Lu, et al. “Ion Beam Analysis of the Annealing Behavior of Helium in Ti Films”. In: *Chin. Phys. Lett.* 26 (2009), p. 042501 (cit. on p. 148).
- [54] M. J. F. Healy, M. Torres, and J. D. Painter. “A software tool enabling the analysis of small lateral features without the use of a micro-beam”. In: *Nucl. Instr. Meth. B* 249 (2006), p. 789 (cit. on p. 76).
- [55] C.P. Hobbs, J.W. McMillan, and D.W. Palmer. “The Effects of Surface Topography in Nuclear Microprobe Rutherford Backscattering Analysis”. In: *Nucl. Instr. Meth. B* 30 (1988), p. 342 (cit. on p. 130).
- [56] W. Hösler and R. Darji. “On the nonlinearity of silicon detectors and the energy calibration in RBS”. In: *Nucl. Instr. Meth. B* 85 (1994), p. 602 (cit. on p. 74).
- [57] C. Jeynes, Z.H. Jafri, R.P. Webb, A.C. Kimber, and M.J. Ashwin. “Accurate RBS Measurements of the Indium Content of InGaAs Thin Films”. In: *Surface and Interface Analysis* 25 (1997), p. 254 (cit. on p. 162).
- [58] J.K. Kim, Y.S. Kim, H.W. Choi, G.D. Kim, H.J. Woo, et al. “Measurement of elastic recoil cross sections of light nuclei by ^{35}Cl using inverse scattering”. In: *Nucl. Instr. Meth. B* 129 (1997), p. 323 (cit. on p. 85).

- [59] A. Kitamura, T. Tamai, A. Taniike, Y. Furuyama, T. Maeda, et al. “Simulation of ERD Spectra for a Surface with a Periodic Roughness”. In: *Nucl. Instr. Meth. B* 134 (1998), p. 98 (cit. on p. 130).
- [60] A.R. Knudson. “Effects of Surface Roughness on Backscattering Spectra”. In: *Nucl. Instr. Meth.* 168 (1980), p. 163 (cit. on p. 130).
- [61] G. Konac, S. Kalbitzer, Ch. Klatt, D. Niemann, and R. Stoll. “Energy Loss and Straggling of H and He Ions of keV energies in Si and C”. In: *Nucl. Instr. Meth. B* 136-138 (1998), p. 159. DOI: [10.1016/S0168-583X\(98\)80016-5](https://doi.org/10.1016/S0168-583X(98)80016-5) (cit. on pp. 31, 99, 123, 168).
- [62] G. Konac, Ch. Klatt, and S. Kalbitzer. “Universal fit formula for electronic stopping of all ions in carbon and silicon”. In: *Nucl. Instr. Meth. B* 146 (1998), p. 106. DOI: [10.1016/S0168-583X\(98\)00453-4](https://doi.org/10.1016/S0168-583X(98)00453-4) (cit. on pp. 31, 99).
- [63] N. Kubota, K. Kondo, K. Ochiai, and T. Nishitani. “Neutron elastic recoil detection for hydrogen isotope analysis in fusion materials”. In: *J. Nucl. Mater.* 367-370 (2007), p. 1596 (cit. on p. 62).
- [64] M.A. Kumakhov and F.F. Komarov. *Energy Loss and Ion Ranges in Solids*. New York, London, Paris: Gordon and Breach Science Publishers, 1981 (cit. on p. 103).
- [65] M. Küstner, W. Eckstein, E. Hechtel, and J. Roth. “Angular Dependence of the Sputtering Yield of Rough Beryllium Surfaces”. In: *J. Nucl. Mater.* 265 (1999), p. 22 (cit. on p. 136).
- [66] J. L’Ecuyer, J.A. Davies, and N. Matsunami. “How accurate are absolute Rutherford backscattering yields”. In: *Nucl. Instr. Meth.* 160 (1979), p. 337 (cit. on pp. 84, 168).
- [67] H. Langhuth, M. Mayer, and S. Lindig. “Layer Morphology Analysis of Sputter-eroded Silicon Gratings using Rutherford Backscattering”. In: *Nucl. Instr. Meth. B* 269 (2011), p. 1811. DOI: [10.1016/j.nimb.2011.05.002](https://doi.org/10.1016/j.nimb.2011.05.002). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0026-EAC7-7>. URL: [https://mam.home.ipp.mpg.de/Langhuth-Nuclear-Instruments-Methods-B-269-\(2011\)-1811.pdf](https://mam.home.ipp.mpg.de/Langhuth-Nuclear-Instruments-Methods-B-269-(2011)-1811.pdf) (cit. on p. iv).
- [68] S.R. Lee and R.R. Hart. “Screened Rutherford backscattering cross sections for heavy ion and low energy backscattering analysis”. In: *Nucl. Instr. Meth. B* 79 (1993), p. 463 (cit. on p. 85).
- [69] J. Liu, Z. Zheng, and W.K. Chu. “Stopping Cross Sections of C, Al, and Si for ^7Li Ions”. In: *Nucl. Instr. Meth. B* 118 (1996), p. 24 (cit. on p. 96).
- [70] A.D. Marwick and P. Sigmund. “Small-Angle Multiple Scattering of Ions in the Screened Coulomb Region”. In: *Nucl. Instr. Meth.* 126 (1975), p. 317 (cit. on p. 116).
- [71] J.W. Mayer and E. Rimini. *Ion Handbook for Material Analysis*. New York, San Francisco, London: Academic Press, 1977 (cit. on pp. 103–105, 113).
- [72] M. Mayer. *SIMNRA User’s Guide*. Tech. rep. IPP 9/113. Garching: Max-Planck-Institut für Plasmaphysik, 1997. URL: <https://mam.home.ipp.mpg.de/Report%20IPP%209-113.pdf> (cit. on pp. iii, 123).

- [73] M. Mayer. “SIMNRA, a Simulation Program for the Analysis of NRA, RBS and ERDA”. In: *Proceedings of the 15th International Conference on the Application of Accelerators in Research and Industry*. Ed. by J. L. Duggan and I.L. Morgan. Vol. 475. AIP Conference Proceedings. Woodbury, New York: American Institute of Physics, 1999, p. 541. DOI: [10.1063/1.59188](https://doi.org/10.1063/1.59188). URL: [https://mam.home.ipp.mpg.de/Mayer-AIP-Conference-Proceedings-475-\(1999\)-541.pdf](https://mam.home.ipp.mpg.de/Mayer-AIP-Conference-Proceedings-475-(1999)-541.pdf) (cit. on pp. iii, 61).
- [74] M. Mayer. “Ion Beam Analysis of Rough Thin Films”. In: *Nucl. Instr. Meth. B* 194 (2002), p. 177. DOI: [10.1016/S0168-583X\(02\)00689-4](https://doi.org/10.1016/S0168-583X(02)00689-4). URL: [https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-194-\(2002\)-177.pdf](https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-194-(2002)-177.pdf) (cit. on pp. iii, 137).
- [75] M. Mayer. “Improved Physics in SIMNRA 7”. In: *Nucl. Instr. Meth. B* 332 (2014), p. 176. DOI: [10.1016/j.nimb.2014.02.056](https://doi.org/10.1016/j.nimb.2014.02.056). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0023-C6B3-9>. URL: [https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-332-\(2014\)-176.pdf](https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-332-(2014)-176.pdf) (cit. on pp. iv, 61).
- [76] M. Mayer. “An expert-assisted system for improving the quality of IBA simulations by SIMNRA”. In: *J. Phys.: Conf. Ser.* 2326 (2022), p. 012007. DOI: [10.1088/1742-6596/2326/1/012007](https://doi.org/10.1088/1742-6596/2326/1/012007). URL: [https://mam.home.ipp.mpg.de/Mayer-Journal-Physics-Conference-Series-2326-\(2022\)-012007.pdf](https://mam.home.ipp.mpg.de/Mayer-Journal-Physics-Conference-Series-2326-(2022)-012007.pdf) (cit. on p. iv).
- [77] M. Mayer, K. Arstila, K. Nordlund, E. Edelmann, and J. Keinonen. “Multiple scattering of MeV ions: Comparison between the analytical theory and Monte-Carlo and molecular dynamics simulations”. In: *Nucl. Instr. Meth. B* 249 (2006), p. 823. DOI: [10.1016/j.nimb.2006.03.148](https://doi.org/10.1016/j.nimb.2006.03.148). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0027-0B39-C>. URL: [https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-249-\(2006\)-823.pdf](https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-249-(2006)-823.pdf) (cit. on pp. iii, 117, 120, 123, 126).
- [78] M. Mayer, K. Arstila, and U. von Toussaint. “Skewness of energy-loss straggling and multiple-scattering energy distributions”. In: *Nucl. Instr. Meth. B* 268 (2010), p. 1744. DOI: [10.1016/j.nimb.2010.02.057](https://doi.org/10.1016/j.nimb.2010.02.057). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0026-F07D-B>. URL: [https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-268-\(2010\)-1744.pdf](https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-268-(2010)-1744.pdf) (cit. on pp. iii, 103, 153).
- [79] M. Mayer, R. Behrisch, P. Andrew, and A.T. Peacock. “Erosion at the vessel walls of JET”. In: *J. Nucl. Mater.* 241-243 (1997), p. 469. DOI: [10.1016/S0022-3115\(97\)80083-9](https://doi.org/10.1016/S0022-3115(97)80083-9) (cit. on p. 134).
- [80] M. Mayer, R. Behrisch, C. Gowers, P. Andrew, and A.T. Peacock. “Change of the Optical reflectivity of Mirror Surfaces Exposed to JET Plasmas”. In: *Diagnostics for Experimental Thermonuclear Fusion Reactors 2*. Ed. by P.E. Stott, G. Gorini, P. Prandoni, and E. Sindoni. New York, London: Plenum Press, 1998, p. 279 (cit. on p. 134).
- [81] M. Mayer, W. Eckstein, H. Langhuth, F. Schiettekatte, and U. von Toussaint. “Computer Simulation of Ion Beam Analysis: Possibilities and Limitations”. In: *Nucl. Instr. Meth. B* 269 (2011), p. 3006. DOI: [10.1016/j.nimb.2011.04.066](https://doi.org/10.1016/j.nimb.2011.04.066). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0026-E993-3>. URL: [https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-269-\(2011\)-3006.pdf](https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-269-(2011)-3006.pdf) (cit. on p. iv).

- [82] M. Mayer, U. von Toussaint, J. Dewalque, O. Dubreuil, C. Henrist, et al. “Rutherford backscattering analysis of porous thin TiO₂ films”. In: *Nucl. Instr. Meth. B* 273 (2012), p. 83. DOI: [10.1016/j.nimb.2011.07.045](https://doi.org/10.1016/j.nimb.2011.07.045). eDoc: <http://hdl.handle.net/11858/00-001M-0000-0026-E4F3-2>. URL: [https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-273-\(2012\)-83.pdf](https://mam.home.ipp.mpg.de/Mayer-Nuclear-Instruments-Methods-B-273-(2012)-83.pdf) (cit. on pp. iv, 154).
- [83] H. Metzner, M. Gossila, and Th. Hahn. “Rutherford Backscattering Spectroscopy of Rough Films: Theoretical Considerations”. In: *Nucl. Instr. Meth. B* 124 (1997), p. 567 (cit. on p. 130).
- [84] H. Metzner, Th. Hahn, M. Gossila, J. Conrad, and J.-H. Bremer. “Rutherford Backscattering Spectroscopy of Rough Films: Experimental Aspects”. In: *Nucl. Instr. Meth. B* 134 (1998), p. 249 (cit. on p. 130).
- [85] P.J. Mohr, B.N. Taylor, and D.B. Newell. “CODATA recommended values of the fundamental physical constants: 2010”. In: *Rev. Mod. Phys.* 84 (2012), p. 1527. DOI: [10.1103/RevModPhys.84.1527](https://doi.org/10.1103/RevModPhys.84.1527) (cit. on p. 81).
- [86] P.J. Mohr, B.N. Taylor, and D.B. Newell. “CODATA Recommended Values of the Fundamental Physical Constants: 2014”. In: *Journal of Physical and Chemical Reference Data* 45 (2016), p. 043102. DOI: [10.1063/1.4954402](https://doi.org/10.1063/1.4954402) (cit. on p. 81).
- [87] J.A. Nelder and R. Mead. In: *Computer Journal* 7 (1965), p. 308 (cit. on p. 49).
- [88] F. Pászti and E. Szilágyi. “Pore structure investigations in porous silicon by ion beam analytical methods”. In: *Vacuum* 50 (1998), p. 451 (cit. on p. 148).
- [89] H. Paul. *Stopping Power for Light Ions: Graphs, Data, Comments and Programs*. <http://www-nds.iaea.org/stopping/>. Last accessed on 2013-09-19 (cit. on p. 96).
- [90] H. Paul, A. Schinner, and P. Sigmund. “Electronic Stopping of Swift Lithium and Carbon Ions”. In: *Nucl. Instr. Meth. B* 164-165 (2000), p. 212 (cit. on p. 96).
- [91] M.G. Payne. “Energy Straggling of Heavy Charged Particles in Thick Absorbers”. In: *Phys. Rev.* 185.2 (1969), p. 611 (cit. on pp. 103, 123).
- [92] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes*. Cambridge, New York: Cambridge University Press, 1988 (cit. on pp. 49–51, 93, 122).
- [93] F. Schiettekatte. “Fast Monte Carlo for ion beam analysis simulations”. In: *Nucl. Instr. Meth. B* 266 (2008), p. 1880. DOI: [10.1016/j.nimb.2007.11.075](https://doi.org/10.1016/j.nimb.2007.11.075) (cit. on p. 120).
- [94] A. Schinner and P. Sigmund. “Expanded PASS stopping code”. In: *Nucl. Instr. Meth. B* 460 (2019), p. 19. DOI: [10.1016/j.nimb.2018.10.047](https://doi.org/10.1016/j.nimb.2018.10.047) (cit. on pp. 32, 168).
- [95] A. Schinner and P. Sigmund. *DPASS*. <https://www.sdu.dk/en/DPASS> (cit. on p. 32).
- [96] H.G. Schlager and W. Eckstein. *The Scattering Integrals - Integration and Accuracy*. Tech. rep. IPP 9/69. Garching: Max-Planck-Institut für Plasmaphysik, 1991 (cit. on p. 87).
- [97] O. Schmelmer, G. Dollinger, C.M. Frey, A. Bergmaier, and S. Karsch. “Energy Straggling of 60 MeV ⁵⁸Ni^{q+} Ions in Thin Carbon Foils and Gases”. In: *Nucl. Instr. Meth. B* 145 (1998), p. 261 (cit. on p. 111).

Bibliography

- [98] V.S. Shorin and A.N. Sosnin. “RBS Spectra for Thin Films with Surface Roughness”. In: *Nucl. Instr. Meth. B* 72 (1992), p. 452 (cit. on p. 130).
- [99] P. Sigmund. *Particle Penetration and Radiation Effects*. Vol. 151. Springer Series in Solid State Sciences. Berlin, Heidelberg: Springer, 2006 (cit. on pp. 120, 123).
- [100] P. Sigmund and A. Schinner. “Binary stopping theory for swift heavy ions”. In: *Eur. Phys. J. D* 12 (2000), p. 425. doi: [10.1007/s100530070004](https://doi.org/10.1007/s100530070004) (cit. on p. 32).
- [101] P. Sigmund and A. Schinner. “Binary theory of electronic stopping”. In: *Nucl. Instr. Meth. B* 195 (2002), p. 64. doi: [10.1016/S0168-583X\(01\)01162-4](https://doi.org/10.1016/S0168-583X(01)01162-4) (cit. on p. 32).
- [102] P. Sigmund and K.B. Winterbon. “Small-Angle Multiple Scattering of Ions in the Screened Coulomb Region”. In: *Nucl. Instr. Meth.* 119 (1974), p. 541 (cit. on p. 116).
- [103] B.G. Skorodumov, I.O. Yatsevich, V.G. Ulanov, E.V. Zhukovska, and O.A. Zhukovsky. “ERD technique with monochromatic neutrons as an effective tool for hydrogen-material systems study”. In: *Nucl. Instr. Meth. B* 85 (1994), p. 803 (cit. on p. 62).
- [104] E. Steinbauer, P. Bauer, and J.P. Biersack. “Monte Carlo Simulation of RBS Spectra: Comparison to Experimental and Empirical Results”. In: *Nucl. Instr. Meth. B* 45 (1990), p. 171 (cit. on pp. 116, 170).
- [105] E. Steinbauer, P. Bauer, M. Geretschlager, G. Bortels, J.P. Biersack, et al. “Energy Resolution of Silicon Detectors: Approaching the Physical Limit”. In: *Nucl. Instr. Meth. B* 85 (1994), p. 642 (cit. on p. 74).
- [106] J.P. Stoquert and T. Szörényi. “Determination of the number and size of inhomogeneities in thin films by ion beam analysis”. In: *Phys. Rev. B* 66 (2002), p. 144108 (cit. on p. 148).
- [107] E. Szilágyi. “Energy Spread in Ion Beam Analysis”. In: *Nucl. Instr. Meth. B* 161-163 (2000), p. 37. doi: [10.1016/S0168-583X\(99\)00671-0](https://doi.org/10.1016/S0168-583X(99)00671-0) (cit. on pp. 102, 120).
- [108] E. Szilágyi. *DEPTH for targets containing multilayers*. <http://www.kfki.hu/~ionhp/doc/prog/mdepth.htm>. Last accessed on 2013-09-19 (cit. on p. 117).
- [109] E. Szilágyi, F. Pászti, and G. Amsel. “Theoretical approximations for depth resolution calculations in IBA methods”. In: *Nucl. Instr. Meth. B* 100 (1995), p. 103. doi: [10.1016/0168-583X\(95\)00186-7](https://doi.org/10.1016/0168-583X(95)00186-7) (cit. on pp. 32, 102–106, 113, 116, 117, 120, 122, 123, 126, 153).
- [110] J.R. Tesmer and M. Nastasi, eds. *Handbook of Modern Ion Beam Materials Analysis*. First. Pittsburgh, Pennsylvania: Materials Research Society, 1995 (cit. on pp. 84, 168).
- [111] D.I. Thwaites. In: *Nucl. Instr. Meth. B* 27 (1987), p. 293 (cit. on p. 100).
- [112] E. Tiesinga, P.J. Mohr, D.B. Newell, and B.N. Taylor. “CODATA recommended values of the fundamental physical constants: 2018”. In: *Rev. Mod. Phys.* 93 (2021), p. 025010. doi: [10.1103/RevModPhys.93.025010](https://doi.org/10.1103/RevModPhys.93.025010) (cit. on p. 81).
- [113] J. Tirira, Y. Serruys, and P. Trocellier. *Forward Recoil Spectrometry*. New York, London: Plenum Press, 1996 (cit. on p. 103).
- [114] M. Tosaki. “Energy-loss straggling caused by the inhomogeneity of target material”. In: *J. Appl. Phys.* 99 (2006), p. 034905 (cit. on p. 148).

- [115] C. Tschalär. “Straggling distributions of extremely large energy losses”. In: *Nucl. Instr. Meth.* 64 (1968), p. 237. doi: [10.1016/0029-554X\(68\)90159-6](https://doi.org/10.1016/0029-554X(68)90159-6) (cit. on pp. 103, 120, 123).
- [116] C. Tschalär. “Straggling distributions of large energy losses”. In: *Nucl. Instr. Meth.* 61 (1968), p. 141. doi: [10.1016/0029-554X\(68\)90535-1](https://doi.org/10.1016/0029-554X(68)90535-1) (cit. on pp. 103, 120, 123).
- [117] C. Tschalär and H.D. Maccabee. “Energy-Straggling Measurements of Heavy Charged Particles in Thick Absorbers”. In: *Phys. Rev. B* 1 (1970), p. 2863 (cit. on pp. 123–125).
- [118] P.V. Vavilov. In: *Soviet Physics J.E.T.P.* 5 (1957), p. 749 (cit. on pp. 103, 122).
- [119] P.F. Velleman. “Definition and Comparison of Robust Nonlinear Data Smoothing Algorithms”. In: *Journal of the American Statistical Association* 75.371 (1980), p. 609. JSTOR: [2287657](https://www.jstor.org/stable/2287657) (cit. on p. 75).
- [120] J. Vorona, J.W. Olness, W. Haeberli, and H.W. Lewis. In: *Phys. Rev.* 116 (1959), p. 1563 (cit. on p. 173).
- [121] M. Wang, G. Audi, A.H. Wapstra, F.G. Kondev, M. MacCormick, et al. “The AME2012 atomic mass evaluation”. In: *Chinese Physics C* 36 (2012), p. 1603 (cit. on p. 81).
- [122] Y. Wang and M. Nastasi, eds. *Handbook of Modern Ion Beam Materials Analysis*. Second. Warrendale, Pennsylvania: Materials Research Society, 2009 (cit. on pp. 4, 168, 169).
- [123] A. Weber and H. Mommsen. “Background in Rutherford Backscattering Spectra: A Simple Formula”. In: *Nucl. Instr. Meth.* 204 (1983), p. 559 (cit. on p. 116).
- [124] A. Weber, H. Mommsen, W. Sarter, and A. Weller. “Double Scattering in Rutherford Backscattering Spectra”. In: *Nucl. Instr. Meth.* 198 (1982), p. 527 (cit. on p. 116).
- [125] L. Wielopolski and R.P. Gardner. “Prediction of the pulse-height spectral distortion caused by the peak pile-up effect”. In: *Nucl. Instr. Meth.* 133 (1976), p. 303. doi: [10.1016/0029-554X\(76\)90623-6](https://doi.org/10.1016/0029-554X(76)90623-6) (cit. on pp. 156, 158).
- [126] M. Wielunski, M. Mayer, R. Behrisch, J. Roth, and B.M.U. Scherzer. “Simultaneous Profiling of Hydrogen and Deuterium by 2.6 MeV ^4He ERDA using a $\Delta E - E$ Telescope Detector”. In: *Nucl. Instr. Meth. B* 122 (1997), p. 113. doi: [10.1016/S0168-583X\(96\)00767-7](https://doi.org/10.1016/S0168-583X(96)00767-7) (cit. on p. 175).
- [127] M. Wüest and P. Bochsler. “Simulation of Ion Backscattering from Rough Surfaces”. In: *Nucl. Instr. Meth. B* 71 (1992), p. 314 (cit. on p. 130).
- [128] Q. Yang, D.J. O’Connor, and Z. Wang. “Empirical Formulae for Energy Loss Straggling of Ions in Matter”. In: *Nucl. Instr. Meth. B* 61 (1991), p. 149. doi: [10.1016/0168-583X\(91\)95454-L](https://doi.org/10.1016/0168-583X(91)95454-L) (cit. on pp. 104, 106, 107, 120, 122, 123, 168).
- [129] I.M. Yesil. “Einfluß der Oberflächenrauigkeit auf ERDA-Tiefen-Profil”. In german. MA thesis. München: Ludwig Maximilian Universität, 1995 (cit. on p. 130).
- [130] I.M. Yesil, W. Assmann, H. Huber, and K.E.G. Löbner. “Simulation of Surface Roughness Effects in ERDA”. In: *Nucl. Instr. Meth. B* 136-138 (1998), p. 623. doi: [10.1016/S0168-583X\(97\)00860-4](https://doi.org/10.1016/S0168-583X(97)00860-4) (cit. on p. 130).

Bibliography

- [131] J.F. Ziegler. *Helium - Stopping Powers and Ranges in All Elements*. Vol. 4. The Stopping and Ranges of Ions in Matter. New York: Pergamon Press, 1977 (cit. on pp. [31](#), [96](#), [97](#), [168](#)).
- [132] J.F. Ziegler. *Stopping Cross-Sections for Energetic Ions in all Elements*. Vol. 5. The Stopping and Ranges of Ions in Matter. New York: Pergamon Press, 1980 (cit. on p. [123](#)).
- [133] J.F. Ziegler. “private communication”. 1997 (cit. on p. [31](#)).
- [134] J.F. Ziegler. “RBS/ERD Simulation Problems: Stopping Powers, Nuclear Reactions and Detector Resolution”. In: *Nucl. Instr. Meth. B* 136-138 (1998), p. 141. DOI: [10.1016/S0168-583X\(97\)00664-2](#) (cit. on p. [96](#)).
- [135] J.F. Ziegler. “SRIM-2003”. In: *Nucl. Instr. Meth. B* 219-220 (2004), p. 1027. DOI: [10.1016/j.nimb.2004.01.208](#) (cit. on pp. [120](#), [123](#), [168](#)).
- [136] J.F. Ziegler, J.P. Biersack, and U. Littmark. *The Stopping and Range of Ions in Solids*. Vol. 1. The Stopping and Ranges of Ions in Matter. New York: Pergamon Press, 1985 (cit. on pp. [31–33](#), [86](#), [97–99](#), [105](#), [123](#), [168](#)).
- [137] J.F. Ziegler and J.M. Manoyan. “The stopping of ions in compounds”. In: *Nucl. Instr. Meth. B* 35 (1988), p. 215. DOI: [10.1016/0168-583X\(88\)90273-X](#) (cit. on pp. [97](#), [98](#), [100](#)).

Index

- χ^2 evaluation, [49](#)
- 32-bit, [2](#)
- 64-bit, [2](#)
- Access keys, [339](#)
- Accuracies
 - Energy loss, [36](#)
 - Straggling width, [36](#)
- ADC offset, [27](#)
- Administrator privileges, *see* Installation, privileges, *see* Registration, privileges
- Adobe
 - Acrobat, [2](#), [60](#)
 - Acrobat Reader DC, [2](#), [60](#)
 - Reader, [2](#), [60](#)
- Andersen, *see* Screening, Andersen, *see* Stopping power, Andersen-Ziegler
- ANSI, *see* Character encoding, ANSI
- Asymmetric Gaussian
 - Definition, [21](#)
 - Fitting function, [348](#)
 - Graphical representation, [22](#)
- Asymmetry, *see* Straggling, skewness
- Atomic data, [81](#)
- AVA file, [13](#)
- Backup file, [59](#)
- backup.xnra, [59](#)
- Biersack, *see* Stopping power, Ziegler-Biersack
- Bohr, *see* Straggling, Bohr
- Bozoian formula, [87](#)
- Bragg's rule, *see* Stopping power, Bragg's rule
- Calculate menu, [48–53](#)
- Calculate Spectrum, [48](#)
- Calculate Spectrum Fast, [48](#)
- Cross Section, [51](#)
- Density, [52](#)
- Exit Angle Beta, [52](#)
- Fit Spectrum, [48](#)
- Kinematics, [51](#)
- Particles*sr, [52](#)
- Stopping, [51](#)
- Subtract Pile-up, [52](#)
- CAM file, [14](#), [27](#), [29](#)
- Canberra, [13](#), [14](#), [27](#), [29](#)
 - ProSpect, [13](#)
- Character encoding
 - ANSI, [12](#), [13](#), [15](#), [25](#)
 - Unicode, [10](#), [12](#), [13](#), [15](#), [25](#)
 - UTF-8, [12](#), [13](#), [15](#), [25](#)
- Chu, *see* Straggling, Chu
- Command line parameters, [76](#)
- Conventions, [1](#)
- Cornell geometry, *see* Geometry, Cornell
- Coulomb barrier, [168](#)
- Cross-section
 - and straggling, [33](#), [90](#), [91](#)
 - calculation, [51](#)
- Cross-section data, [84–87](#)
 - adding new data, [67](#)
 - CRSDA.DAT file, [48](#)
 - IBANDL, [46–48](#), [67](#)
 - integrated, [90](#)
 - Mott scattering, [60](#), [88](#), [177](#)
 - non-Rutherford cross-section, [87](#)
 - R33 file format, [46](#), [67](#), [341](#)
 - RTR file format, [47](#), [48](#)
 - Rutherford cross-section, [84–87](#)
 - high energy deviation, [87](#)

- recoil cross-section, 84
 - scattering cross-section, 84
 - screening, 33, 84–87
- Cutoff energy, 36
- Data exchange, 63
 - Excel, 63
 - Origin, 63
 - RUMP, 63
 - WiNDF, 65
- Dead layer, *see* Detector, dead layer
- Dead time, 155
- Decimal separator, 10, 12, 13
- Density conversion, 52
- Depth profile, 41
- Detector
 - dead layer, 74
 - electrostatic detector, 24, 101
 - Delta-E/E, 24
 - energy resolution, 101
 - energy calibration, 20, 74
 - detector nonlinearity, 20, 74
 - different ion species, 21, 74
 - non-linear, 20, 74
 - quadratic, 20, 74
 - energy resolution, 21, 101
 - different ion species, 21
 - free flight path, 24, 101
 - plasma effect, 74
 - pulse height defect, 74
 - resolution, *see* Detector, energy resolution, *see* Detector, time resolution
 - semiconductor detector, 74
 - sensitivity, 24
 - solid angle, 21
 - solid state detector, 21, 24, 74
 - material, 24
 - thickness, 24
 - time resolution, 24, 101
 - time-of-flight detector, 21, 24, 101
 - energy resolution, 101
 - free flight path, 24, 101
 - time resolution, 24, 101
 - top electrode, 74
 - transmission detector, 24
 - type, 24
- Detector geometry, 25
- Doolittle, *see* Energy loss evaluation, Doolittle
- DPASS, *see* Stopping power, DPASS
- Dual scattering, 30
 - and surface roughness, 37
- Edit menu, 19
 - Copy data, 19
 - Copy page, 19
- Electronic energy loss straggling
 - skewness, 122
- Energy
 - of incident beam, 20
- Energy calibration, *see* Detector, energy calibration
- Energy loss
 - in foil, 51
 - in layer, 51
 - in target, 51
 - in window, 51
- Energy loss evaluation, 93–94
 - Doolittle, 93
 - Runge-Kutta, 93
- Energy spread of incident beam, *see* Incident beam, incident beam
- Exit angle, 20, 22, 52, 166
- Experimental data, *see* Spectrum data
- Expert system, 56, 165–169
 - Coulomb barrier, 168
 - formal correctness of parameters, 166
 - geometry, 166
 - limitations, 169
 - metadata, 166
 - multiple scattering, 169
 - notifications, 165
 - plural scattering, 169
 - screening, 168
 - stopping power, 168
 - straggling, 168
- FAST ComTec, 15

- File format
 - idf, 11, 12
 - nra, 9, 11, 76
 - xnra, 9, 11, 76, 78
- File menu, 11–18
 - Exit, 18
 - New, 11
 - Open, 9, 11
 - IDF, 11
 - nra, 11
 - xnra, 11
 - Print, 18
 - Read Spectrum Data, 12
 - ASCII, 12
 - ASCII (One column), 12
 - Canberra AVA, 13
 - Canberra CAM, 14
 - Energy/Counts, 14
 - FAST MPA, 15
 - IAEA SPE, 15
 - IPP, 15
 - ISI, 15
 - MCERD, 15
 - RUMP spectrum, 15
 - Swallow, 15
 - User, 16
 - XNRA and IDF, 16
 - RUMP
 - Read RBS File, 17
 - Read Sample Description File, 18
 - Write Sample Description File, 18
 - Save, 11
 - Save as, 11
 - IDF, 12
 - xnra, 11
 - Write Spectrum Data, 17
- Fit error, 50
- Fit spectrum, 48
 - accuracy, 50
 - calculate fit error, 50
 - fit error, 50
 - max iterations, 50
- Gamma distribution, 130
- γ -ray spectrum, 52
- Gaussian
 - asymmetric
 - Definition, 21
 - Fitting function, 348
 - Graphical representation, 22
- Genie-2000, 2, 14
- Geometrical straggling, 25, 113
- Geometry, 20, 166
 - Cornell, 20, 52, 53, 114
 - General, 114
 - IBM, 20, 52, 53, 113, 137
- Help menu, 60–61
 - About, 60
 - Home Page, 60
 - IBANDL Cross-Sections, 60
 - Mott Cross-Sections, 60
 - Register, 61
 - User's Guide, 60
- High dpi devices, 2, 10
- IBA data furnace, 65
- IBANDL, 46–48, 67
- IBM geometry, *see* Geometry, IBM
- IDF file format, *see* File format, idf
- Incident angle, 20, 22, 166
- Incident beam, 21
 - asymmetric Gaussian distribution, 21
 - charge state, 21
 - energy, 20
 - energy spread, 21, 22
 - Gaussian distribution, 21
 - incident particle, 20
 - monoenergetic, 21
 - number of particles, 21
- Infotip handler, *see* Shell extensions, Infotip handler
- Installation, 2
 - privileges, 2
 - registration, 3
 - system requirements, 2
 - Windows version, 2
- Keyboard shortcuts, 339

- Kinematics, 82–83
 - calculation, 51
 - elastic scattering, 82
 - backscattering, 82
 - recoils, 82
 - nuclear reactions, 82
- L'Ecuyer, *see* Screening, L'Ecuyer
- Layer
 - manipulation, 40
 - porosity, 40, 44, 148–153
- Layer roughness, *see* Surface roughness, layer roughness
- License agreement, xii
- Live time, 27, 52, 155, 166
- Live time correction, 27, 155, 166
- Logfile, 34
- Max iterations, 50
- MCERD, 15
- Metadata, 166
- Mott scattering, *see also* MottCalc, *see* Cross-section data, Mott scattering
- MottCalc, 60, 88, 177
- MPA file, 15
- Multiple scattering, 30, 116–118, 169
 - model, 32
 - skewness, 123
- n-ERD, n-ERDA, *see* Neutron-ERD
- NDF, 65
- NERDA, *see* Neutron-ERD
- Neutron-ERD, 62
- Neutron-induced reactions, 62
- nra file format, *see* File format, nra
- Nuclear energy loss straggling, *see* Straggling, nuclear energy loss straggling
- Nuclear stopping, 99
 - universal potential, 99
- OLE automation, 76, 178–332
 - data types, 191
 - methods, 179
 - objects, 178
 - startup, 178
- Options menu, 59
 - Create List of Cross-Sections, 59
 - Preferences, 59
 - directories, 59
 - print, 59
 - saving, 59
 - spectrum data, 59
- Parameters, *see* Command line parameters
- Particles*sr, 21
 - calculation, 52
- Payment, xi
- Payne, 103
- PIGE, 61–62, 76
- Pile-up, 27, 37, 155–162, 166, 168
 - accurate model, 37, 156–160
 - calculation from experimental data, 52
 - fast model, 37, 156, 162
 - pile-up rejector, 29, 52, 160
 - pair resolution time, 29, 52, 160
 - subtraction, 52
- Pile-up rejector, *see* Pile-up, pile-up rejector
- PIXE, 52
- Plasma effect, *see* Detector, plasma effect
- Plot
 - context menu, 58
 - pan, 58
 - zoom in, 58
 - zoom out, 58
- Plot menu, 57
 - Autoscaling, 57
 - Delete Experimental Data, 58
 - Delete Simulated Data, 58
 - Legend, 57
 - Outside of chart, 57
 - Rescale x-Axis, 57
 - Rescale y-Axis, 57
 - Show Spectrum, 57
 - Show Experimental Spectrum, 57
 - Show Simulated Element Spectra, 57
 - Show Simulated Isotope Spectra, 57
 - Show Simulated Spectrum, 57

- Show Smoothed Experimental Spectrum, [57](#)
- Spectrum, [57](#)
- Unzoom, [57](#)
- x-Axis, [57](#)
- y-Axis, [57](#)
- Plural scattering, [30](#), [116](#), [118](#), [169](#)
- Porosity, [40](#), [44](#), *see* Layer, porosity
- Privileges, *see* Installation, privileges, *see* Registration, privileges
- Property handler, *see* Shell extensions, Property handler
- ProSpect software, [13](#)
- Publications, *see* SIMNRA, publications
- Pulse height defect, *see* Detector, pulse height defect
- Pulse rise time, *see* Rise time
- R33 file format, *see* Cross-section data, R33 file format
- Reactions menu, [46](#)
- Real time, [27](#), [52](#), [155](#), [166](#)
- Regional settings, [9](#), [10](#)
- Registration, [xi](#), [3](#), [61](#)
 - privileges, [3](#), [61](#)
 - status, [61](#)
- Rise time, [27](#), [52](#)
- Roughness, *see* Surface roughness
- RTR file format, *see* Cross-section data, RTR file format
- RUMP, [63](#), [79](#), [93](#)
 - RBS file, [15](#), [17](#), [63](#)
 - sample description file, [18](#), [63](#)
 - Swallow command, [15](#)
- Runge-Kutta, [93](#)
- Rutherford cross-section, *see* Cross-section data, Rutherford cross-section
- Scattering angle, [20](#), [22](#), [166](#)
- Scattering kinematics, *see* Kinematics
- Screening, [33](#), [84–87](#), [168](#)
 - Andersen, [33](#), [85](#)
 - L’Ecuyer, [33](#), [84](#)
 - None, [33](#)
 - Universal, [33](#)
- Semiconductor detector
 - see* Detector, semiconductor detector, [358](#)
- Setup menu, [20–29](#)
 - Calculation, [30](#)
 - Experiment, [20](#)
 - Experiment: More Options
 - Detector Geometry, [25](#)
 - Detector Type, [24](#)
 - Live Time and Pile-up, [27](#)
- Shell extensions, [333](#)
 - Property handler, [336](#)
 - Thumbnail provider, [333](#)
- SIMNRA
 - publications, [iii](#)
 - reference, [iii](#)
 - version history, [xiii](#)
- Skewness, [120](#), *see* Straggling, skewness
- Smoothing, [75](#)
- Solid state detector, *see* Detector, solid state detector
- SPE file, [15](#)
- Spectrum data, [12](#)
 - ASCII, [12](#)
 - ASCII (One column), [12](#)
 - AVA file, [13](#)
 - CAM file, [14](#)
 - Canberra
 - AVA file, [13](#)
 - CAM file, [14](#)
 - Energy/Counts, [14](#)
 - FAST
 - MPA file, [15](#)
 - IAEA
 - SPE file, [15](#)
 - IPP, [15](#)
 - ISI, [15](#)
 - MCERD, [15](#)
 - MPA file, [15](#)
 - reading, [12](#)
 - RUMP, [15](#), [17](#)
 - SPE file, [15](#)
 - Swallow, [15](#)

- user defined, 16, 66
 - writing, 17
 - XNRA and IDF, 16
- SRIM, *see* Stopping power, SRIM
- Statusbar, 10
- Stepwidth
 - fixed, 35
 - incident ions, 34
 - outgoing ions, 35
- Stepwidths, 34
- Stopping power, 96–100, 168
 - Andersen-Ziegler, 31, 32, 96
 - heavy ions, 97
 - helium, 96
 - hydrogen, 96
 - Bragg's rule, 40, 100
 - calculation, 51
 - cores and bonds model, 100
 - correction factor, 40
 - DPASS, 32, 177
 - electronic, 31
 - in compounds, 100
 - in element, 51
 - in layer, 51
 - KKK stopping, 31, 99
 - nuclear, 32, 99
 - SRIM, 2, 3, 31, 59, 71, 99
 - download, 71
 - installation, 71
 - program directory, 59, 71
 - trouble shooting, 71
 - SRIM 97, 31, 99
 - User defined, 32, 73
 - Ziegler-Biersack, 31, 97
 - heavy ions, 98
 - helium, 98
 - hydrogen, 97
- Straggling, 102–114, 168
 - and cross-section, 33, 90, 91
 - asymmetric Gaussian, 32
 - Bohr theory, 32, 103, 104
 - charge state fluctuations, 104–106
 - Chu theory, 32, 104
 - electronic energy loss, 102–111
 - accuracy of, 106
 - step width control, 106
- Gaussian, 32
- geometrical, 25
- geometrical straggling, 113
- in compounds, 113
- multiple scattering, 30, 116–118
 - model, 32
- non-statistical broadening, 103
- nuclear energy loss straggling, 111
 - Bohr theory, 111
- Payne, 103
- plural scattering, 30, 116, 118
- shape, 32
- skewness, 120
 - asymmetry parameter, 122
 - electronic energy loss straggling, 122
- Symon, 103
- Tschalär, 103
- Vavilov theory, 103
- Yang theory, 32, 104–106
- Substrate roughness, *see* Surface roughness,
 - substrate roughness
- Surface roughness, 130–147
 - and dual scattering, 37
 - layer roughness, 40, 41, 130–134, 145
 - number of steps, 37
 - parameter for calculation, 37
 - rough film, *see* Surface roughness, layer
 - roughness
 - rough substrate, *see* Surface roughness,
 - substrate roughness
 - substrate roughness, 40, 41, 134–147
- Symon, *see* Straggling, Symon
- Target menu, 39–45
 - Foil, 44
 - Target, 39
 - Window, 44
- Thumbnail provider, *see* Shell extensions,
 - Thumbnail provider
- Time-of-flight detector, *see* Detector, time-
 - of-flight detector
- Toolbar, 9

- Tools menu, [54](#)
 - χ^2 evaluation, [55](#)
 - check input, [56](#)
 - data reader, [54](#)
 - integrate spectrum, [54](#)
 - nearest elements, [54](#)
 - resize spectrum, [56](#)
 - smoothed experimental spectrum, [55](#)
- Top electrode, *see* Detector, top electrode
- TRIM, [31](#), *see* SRIM
- Tschalär, *see* Straggling, Tschalär
- Two-piece normal distribution, [120](#)
 - Definition, [21](#)
 - Fitting function, [348](#)
 - Graphical representation, [22](#)
- Unicode, *see* Character encoding, Unicode
- Uninstallation, [7](#)
- Universal potential, [99](#)
- User.dll, [66](#)
- UTF-8, *see* Character encoding, UTF-8
- Vavilov, *see* Straggling, Vavilov
- Version, [ii](#)
 - 32-bit, [2](#)
 - 64-bit, [2](#)
- Version history, *see* SIMNRA, version history
- Weighting of cross-sections by straggling, [33](#)
 - accurate, [91](#)
 - fast, [90](#)
- WiNDF, [65](#)
- Windows version, *see* Installation, Windows version
- X-ray spectrum, [52](#)
- xnra file format, *see* File format, xnra
- Yang, *see* Straggling, Yang
- Ziegler, *see* Stopping power, Andersen-Ziegler, *see* Stopping power, Ziegler-Biersack