



EUROPEAN  
COMMISSION

Community Research



# CINCH-II

(Project Number: 605173)

## DELIVERABLE D3.2

### Computers in Education Type of Exercises

Lead Beneficiary: **UiO**

Due date: 30/05/2015

Released on: 20/10/2015

<b>Authors:</b>	Jon Petter Omtvedt	
<b>For the Lead Beneficiary</b>	<b>Reviewed by Workpackage Leader</b>	<b>Approved by Coordinator</b>
<b>Jon Petter Omtvedt</b>	<b>Jon Petter Omtvedt</b>	<b>Jan John</b>
<i>Jon Petter Omtvedt</i>	<i>Jon Petter Omtvedt</i>	<i>Jan John</i>

Start date of project: **01/06/2013**

Duration: **36 Months**

Project Coordinator:

Jan John

Project Coordinator Organisation:

CTU in Prague

VERSION: 1.2

<b>Project co-funded by the European Commission under the Euratom Research and Training Programme on Nuclear Energy within the Seventh Framework Programme</b>	
<b>Dissemination Level</b>	
<b>PU</b>	Public

**Version control table**

Version number	Date of issue	Author(s)	Brief description of changes made
1.0	5 <sup>th</sup> Oct. 2015	Jon Petter Omtvedt	First draft
1.1	19 <sup>th</sup> Oct. 2015	Jon Petter Omtvedt	Copies of relevant NucWik pages inserted.
1.2	20 <sup>th</sup> Oct. 2015	Jon Petter Omtvedt	Final draft
1.3	20 <sup>th</sup> Oct. 2015	Jan John	Reviewed by the coordinator

**Relevance**

This deliverable contributes to the following Work-Packages and Tasks:

ALL

## WP 1

Task 1.1  Task 1.2  Task 1.3  Task 1.4

## WP 2

Task 2.1  Task 2.2  Task 2.3  Task 2.4

## WP 3

Task 3.1  Task 3.2  Task 3.3  Task 3.4  Task 3.5

## WP 4

Task 4.1  Task 4.2  Task 4.3  Task 4.4

## WP 5

Task 5.1  Task 5.2  Task 5.3  Task 5.4

**Project information**

Project full title:	Cooperation in education and training In Nuclear CHEmistry
Acronym:	CINCH-II
Funding scheme:	Coordination Action
ECGA number:	605173
Programme and call	FP7 EURATOM, FP7-Fission-2013-5.1.1
Coordinator:	Jan John
EC Project Officer:	Georges Van Goethem
Start date – End date:	01/06/13 – 31/05/16 i.e. 36 months
Coordinator contact:	+420 224 358 228, <a href="mailto:jan.john@fjfi.cvut.cz">jan.john@fjfi.cvut.cz</a>
Administrative contact:	+420 245 008 599, <a href="mailto:cinch@evalion.cz">cinch@evalion.cz</a>
Online contacts:	<a href="http://www.cinch-project.eu/">http://www.cinch-project.eu/</a>

**Copyright**

The document is proprietary of the CINCH-II consortium members. No copying or distributing, in any form or by any means, is allowed without the prior written agreement of the owner of the property rights. This document reflects only the authors' view. The European Community is not liable for any use that may be made of the information contained herein.

## **EXECUTIVE SUMMARY**

Three Computing in Science Education (CSE) type of exercises have been developed to a stage where they are ready to be used in the classroom. The exercises illustrate different methods where modern computing tools can benefit and enhance traditional teaching, in some cases making it possible to allow the students to tackle problems which would be absolutely impossible with traditional methods.

Incorporating such modern tools into NRC teaching might also enhance NRC's image and show the students that it really is part of modern science.

The three exercises are openly available on the wiki-service NucWik, and will remain so also after the project is finished.

## CONTENT

<b>1</b>	<b>INTRODUCTION.....</b>	<b>5</b>
1.1	WHAT IS COMPUTING IN SCIENCE EDUCATION (CSE)?.....	6
1.2	PURPOSE OF NUCWIK CSE EXERCISES.....	7
<b>2</b>	<b>DESCRIPTION OF DEVELOPED CSE EXERCISES.....</b>	<b>9</b>
<b>3</b>	<b>CONCLUSIONS .....</b>	<b>11</b>

## 1 INTRODUCTION

This deliverable – Deliverable 3.2 - to the CINCH-II project is the result of work performed in Task 3.3: "Computers in Education" type of Exercises. It is one of five tasks in Work Package 3, and like Task 3.1 in WP 3 has solely been done at the University of Oslo (UiO). (Task 3.3, 3.4, and 3.5 is performed in collaboration with other CINCH-II partners. UiO is WP 3 leader.)

The original description for Task 3.3 was:

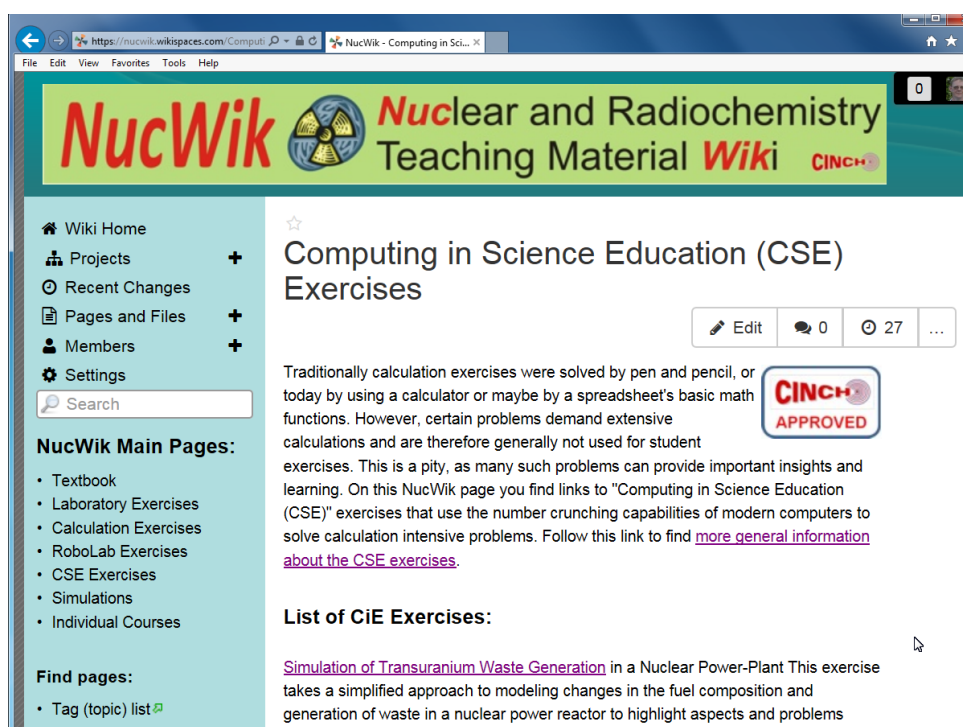
"For the "Computers in Education" approach to work it is important that the more advanced courses will use (and benefit) from the concept. I.e. more complex calculation exercises should be included in the course, which require computer programming to be solved. Therefore, suitable problem solving sets suitable for the "Computers in Education" concept will be developed for use in nuclear chemistry courses. The fuel inventory as function of time in a nuclear reactor would be a typical example."

Deliverable 3.2 is the only and final deliverable from Task 3.3 and is the termination of the work on this task.

Deliverable 3.2 was originally scheduled to be finished by project month 24 (May 2015), but due to personnel issues at UiO the work was delayed. The work was finished in project month 29 (October 2015). The CINCH project board was informed about the delay and accepted this in the Project and Board Meeting #4 in Helsinki, Finland in May 2015.

Although the concept described and developed in this task originally was named "Computers in Education", it will in this report be described more appropriately as "Computing in Science Education" (CSE), as this is more in sync with what is commonly used elsewhere.

The complete work constituting Deliverable 3.2 and Task 3.3 is openly available from the Wiki-server "NucWik", which also was set up as a part of the CINCH-II project. See Task 3.1 "Wiki for sharing and developing teaching material" and Deliverable 3.1 for details. Task 3.1 was finalized in project month 12 (May 2014)) and the service has been openly available since then. NucWik is running on the commercial server "wikispaces" (see report on Deliverable 3.1 for details). All the work constituting Deliverable 3.2 can be accessed from main-menu item "CSE



The screenshot shows a web browser window displaying the NucWik website. The header includes the NucWik logo and the text "Nuclear and Radiochemistry Teaching Material Wiki CINCH+". The main content area is titled "Computing in Science Education (CSE) Exercises" and features a "CINCH+ APPROVED" badge. The page text describes traditional calculation exercises and introduces CSE exercises that use modern computers. A "List of CiE Exercises:" section is visible, with the first entry being "Simulation of Transuranium Waste Generation in a Nuclear Power-Plant". The left sidebar contains navigation options like "Wiki Home", "Projects", "Recent Changes", "Pages and Files", "Members", and "Settings".

Exercises" at <http://NucWik.wikispaces.com>.

A screen dump of the CSE Exercises main page (as per October 2015) is copied her for your convenience. However, as with most internet sites, the content is dynamic and the reader is recommended to visit the actual site for the most up-to-date version.

This also underlines the fact that although Deliverable 3.2 of Task 3.3 is finalized with respect to the CINCH-II project, the work on building CSE exercises and extending the content of the exercises is on-going and will continue past CINCH-II.

Copies of the NucWik pages describing the three CSE exercises developed for CINCH-II are added to the end of this report. However, as already remarked above any copy of a wikipage becomes obsolete as soon as it is printed out. Serious users should always get the most up-to-date material from NucWik directly.

## 1.1 What is Computing in Science Education (CSE)?

The following is excerpts from the NucWik page "General Description of Computing in Science Education (CSE) exercises" (as of October 2015).

Traditionally calculation exercises were solved by pen and pencil, or today by using a calculator or maybe by a spreadsheet's basic math functions. However, certain problems demand extensive calculations and are therefore generally not used for student exercises. This is a pity, as many such problems can provide important insights and learning. On this NucWik page you find links to "Computing in Science Education (CSE)" exercises that use the number crunching capabilities of modern computers to solve calculation intensive problems. Thus, students and teachers are provided the opportunity to work with more complex problems out of reach of traditional classroom methods.

The concept of Computing in Science Education (CSE) has emerged steadily and in parallel with the introduction of e-learning and computers in the classroom. It might not always be performed under the CSE label or as such a definite concept, but it is obvious that harnessing the computing capabilities of modern computers enables teachers to include problems in student exercises that in former days were restricted to research teams using state-of-the-art computers not available to everybody.

For NucWik and the CINCH project the inspiration to include CSE exercises was provided by a project at the Faculty of Mathematics and Natural Sciences at the University of Oslo (UiO)<sup>1</sup>. The project is of course appropriately named "CSE - Computing in Science Education". The goal is described as:

"The goal of the CSE-project is to include computing as a natural tool for all science and engineering students from the first semester of their undergraduate studies. Not as a substitute for more traditional approaches, but as an extension of the classical toolbox."

---

<sup>1</sup> <http://www.mn.uio.no/english/about/collaboration/cse/> link is correct as per October 2015

The motivation for using CSE in teaching is well described at the project's UiO web-page and coincide with the CINCH-project motivation for including such exercises here at NucWik. The motivation is copied here for your convenience:

*"Computer simulation has become such a fundamental tool in research and industry that it should be introduced to students in the first math and science courses. Surprisingly, the use of computers to solve mathematical problems still has little impact on university education around the world, particularly on bachelor level. Given today's dominance of numerical simulations among professionals, we think it is paramount to integrate numerical tools at all levels in the education system.*

*Our focus is to teach students how they can attack mathematical problems by using numerical methods and programming, and how they can explore the mathematical models through experimenting with programs. The pedagogical benefits are three-fold:*

- *Programming helps to increase the "understanding" of the mathematical methods*
- *Simulation with aid of programmes moves the attention from possible intricate algebra with pen and paper to a greater focus on problem formulation and physical behaviour of systems, i.e., "physical insight"*
- *Numerical solutions allow much more realistic and inspiring problems to be addressed, and give students a taste of what research is about*

*A particular achievement of the CSE project in Oslo is that we have managed to implement the computer-based methods by modifying existing science courses, in contrast to the more common approach where modernization is taken care of in new "computational science" student programs with separate courses.*

*To summarize, CSE is about bringing the modern way of working with mathematical models from the professionals to the students, and the beginning students in particular."*

It is clear that at the Faculty of Mathematics and Natural Sciences at UiO a substantial effort has been done to include CSE into teaching at all levels and a large number of courses. Please refer to the UiO CSE web-pages for more details. The UiO working group: Knut Mørken, Nina Sasaki Aanesen, Lars Oswald Dahl, Hugo Lewi Hammer, Terje Brinck Løyning, Anders Malthe-Sørenssen, Elisabeth Nøst, Ingve Simonsen, Jon Eivind Vatne and Tone Skramstad have published a report on how to introduce CSE as an integral part of first degree teaching (date: 1/2/2011. English edition: 15/6/2011). You might find the report useful even for less ambitious CSE projects.

## **1.2 Purpose of NucWik CSE exercises**

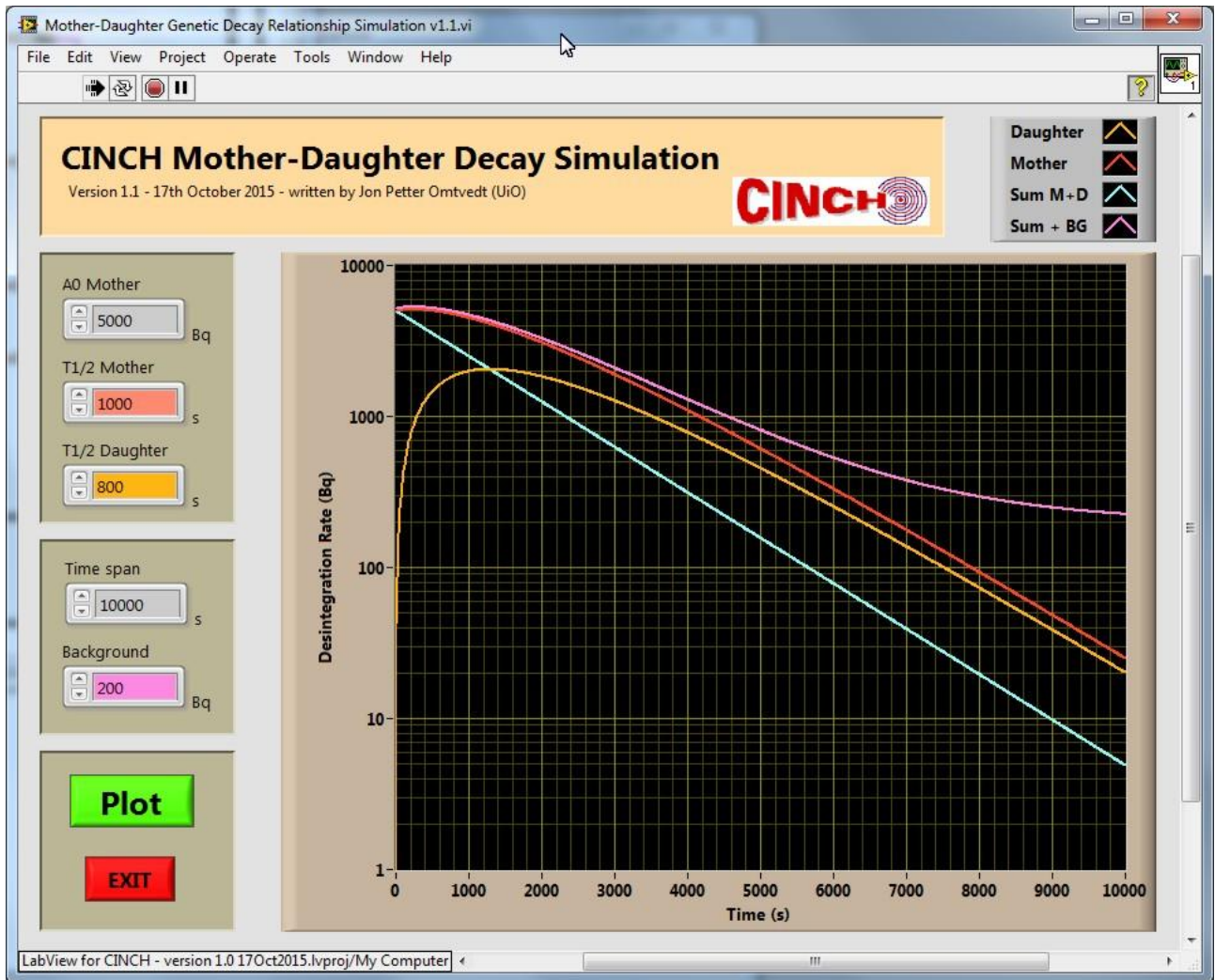
Most universities have not gone to such lengths as UiO to implement CSE, but this does not exclude individual teachers from using CSE in their courses. It is the hope of the CINCH consortium (who set up these NucWik pages) that the CSE exercises provided here as examples can serve as inspiration and help for further exploring CSE in your own teaching.

Students with extensive training in numerical problem solving (e.g. numerical solutions of differential equations, integration techniques, linear algebra methods, root finding and interpolation methods, processing of sound and images, etc.) can of course tackle really



advanced problems. The exercises provided by the CINCH consortium and listed on NucWik is less complex and thus more generally applicable. That said, anyone who has developed more complex exercises is encouraged to share their ideas and material here on NucWik.

Below is a screen shot of one of the programs developed to test and develop the concept behind the CSE exercises delivered here. The program is available from NucWik both as computer code and as an executable program.





## 2 DESCRIPTION OF DEVELOPED CSE EXERCISES

Three exercises were developed under Task 3.3 and constitute Deliverable 3.2:

- **Simulation of Transuranium Waste Generation in a Nuclear Power-Plant:** This exercise takes a simplified approach to modelling changes in the fuel composition and generation of waste in a nuclear power reactor to highlight aspects and problems related to long-lived nuclear waste. The exercise is particularly useful for teaching waste composition and generation as a function of original fuel composition, including fuels containing thorium and plutonium.
- **Simulation of Radionuclide Genetic Dependency (Mother-Daughter relationship):** This exercise calculates the change in radioactivity in a mother-daughter relationship as a function of time, using both an analytic solution (formula) and by numerically solving differential equations. In particular, this exercise is suitable for learning about radionuclide generators.
- **Simulating a Gamma-ray Detector:** A simulated gamma-detector can be programmed to only absorb gamma-rays in selected ways. For example one can choose that the detector does not measure events that are absorbed by the Compton effect. In this way the various ways gamma-rays interact with matter (the detector) and the inherit properties of the detector can be controlled and studied with much greater control than in real life. This exercise is particularly suitable as preparation for hands-on laboratory exercises where real gamma-ray detectors will be used (e.g. HPGe or NaI detectors).

The three exercises exemplify different ways CSE can be used to enhance NRC teaching:

1. The first exercise explores the computer's capability to repeatedly and very quickly do millions of simple calculations. In this way an extremely complex problem can be investigated fairly accurately in a way which is simply not possible with simpler methods.
2. The second exercise illustrates how numerical solutions to differential equations can be used to handle real problems (mother-daughter relationship). In the selected case there is also a relatively straight-forward analytical solution available. This enables the student (and the teacher) to optionally use the analytical solution (and only take advantage of the convenient plotting capabilities of modern computer tools to present the resulting graphs). It is up to the teacher to select which approach serves the sought after learning outcomes best.
3. The third exercise demonstrates another advantage of using computers: The possibility to simulate a complex real process in order to be able to study it more carefully than would be possible in real life. The case selected is the detection of gamma radiation, which is absorbed in matter in a rather complicated way. This implies that measured gamma-ray spectra are shaped according to these absorption processes. In real life one can only observe the sum of the absorption processes, but in a simulation the processes can be switched on or off at will. In that way one can study the processes one by one, without disturbance from the other processes. E.g. one can simulate a measurement of a spectrum from 662 keV  $^{137}\text{Cs}$  gamma-rays without the Compton effect distorting the spectrum. This cannot be done with a real detector, but is simple to do in a simulated detector.

The CSE exercises will not be described in detail here, since detailed descriptions are available

from NucWik. However copies of the main page for each exercise, including the main student and teacher guides (if available) are added as appendixes to this report. As mentioned in the introduction, a wiki document is dynamic and "alive". Thus, content will be updated and changed as usage provides feedback and new ideas (and error correction, of course). If you want to use these exercises for your teaching you should therefore always fetch the most up-to-date versions from NucWik. All teachers are also encouraged to provide feedback to other teachers through the NucWik comment and feedback tools.

The appendixes are as follows:

Appendix 1.1: CSE Exercise – Simulation of Transuranium Waste Generation

Appendix 1.2: General introduction and student guide

Appendix 1.3: Teacher guide

Appendix 2.1: CSE Exercise - Simulation of Radionuclide Genetic-Dependency

Appendix 2.2: Student guide

Appendix 3.1: CSE Exercise – Simulating a Gamma-ray Detector

Appendix 3.2: Student guide on how to use the simulation program

Appendix 3.4: Student guide on how to write the program

### 3 CONCLUSIONS

The promised delivery (Deliverable 3.2) has been met and three ready to use CSE exercises were developed. The exercises can be used in a variety of ways and the level of complexity varied to suit the student group and available time.

The work provided here has been inspired by the great work at the University of Oslo on trying to bring modern research methods and tools into the classroom. It is our hope that this will add to make NRC teaching more "modern" and in sync with the emerging computer-literate students now sweeping over our universities. In this respect, the teachers have often more to learn than their students.

# Computing in Science Education (CSE) Exercises

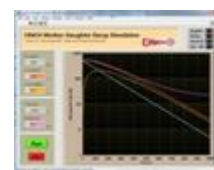
Traditionally calculation exercises were solved by pen and pencil, or today by using a calculator or maybe by a spreadsheet's basic math functions. However, certain problems demand extensive calculations and are therefore generally not used for student exercises. This is a pity, as many such problems can provide important insights and learning. On this NucWik page you find links to "Computing in Science Education (CSE)" exercises that use the number crunching capabilities of modern computers to solve calculation intensive problems. Follow this link to find [more general information about the CSE exercises](#).



## List of CiE Exercises:

[Simulation of Transuranium Waste Generation](#) in a Nuclear Power-Plant This exercise takes a simplified approach to modelling changes in the fuel composition and generation of waste in a nuclear power reactor to highlight aspects and problems related to long-lived nuclear waste. The exercise is particularly useful for teaching waste composition and generation as a function of original fuel composition, including fuels containing thorium and plutonium.

[Simulation of Radionuclide Genetic Dependency \(Mother-Daughter relationship\)](#). This exercise calculates the change in radioactivity in a mother-daughter relationship as a function of time, using both an analytic solution (formula) and by numerically solving differential equations. In particular, this exercise is suitable for learning about radionuclide generators.




[Simulating a Gamma-ray Detector](#). A simulated gamma-detector can be programmed to only absorb gamma-rays in selected ways. For example one can choose that the detector do not measure events that are absorbed by the Compton effect. In this way the various ways gamma-rays interact with matter (the detector) and the inherit properties of the detector can be controlled and studied with much greater control than in real life. This exercise is particularly suitable as preparation for hands-on laboratory exercises where real gamma-ray detectors will be used (e.g. HPGe or NaI detectors).

## Feedback

As always, we would much like to receive comments, suggestions, corrections, etc. Please provide such on the associated comment pages or send an e-mail to [NucWik-post@kjemi.uio.no](mailto:NucWik-post@kjemi.uio.no).

Likewise if you have developed your own CSE exercises - please share your work with the NucWik community!

Add Discussion

Contributions to <https://nucwik.wikispaces.com/> are licensed under a [Creative Commons Attribution Share-Alike 3.0 License](#). 

Portions not contributed by visitors are Copyright 2015 Tangient LLC

# CSE Exercise - Simulation of Transuranium Waste Generation

## Summary

The purpose of this exercise is to study how transuranium nuclear waste is generated in a nuclear reactor. For this purpose a computer program will be written that simulates the nuclear reactions and disintegrations inside a nuclear-reactor core loaded with a specific fuel mixture. To keep the project manageable and relatively small we will introduce simplifications that make it possible to write the program and perform the simulations within a limited amount of time (about one day). Nevertheless the simulation will provide important understanding and insight into the dynamics of a running nuclear reactor. The main simplification used is that only thermal neutrons are present in our simulated reactor.



Follow this link for a general description of the concept of [Computing in Science Education \(CSE\) exercises](#).

## Learning Outcome

Performing this exercise should teach the student:

- A good understanding of the dynamics inside a reactor core (how the decay and in-growth of all the different nuclei is interlinked and depend on each other).
- A fundamental understanding of how the neutron flux will change the waste composition.
- A good understanding of how the waste composition changes as a function of time (both during irradiation and after).
- A good understanding of how the fuel composition will dictate the waste composition.
- Develop a more intimate feel for the basic workings of nuclear reactor.
- In addition, the student will learn how complex technology can be investigated by computer simulations.

## Limitations

Many aspects of constructing and operating a nuclear reactor is not considered in this exercise. In particular, the importance of considering the effect neutron energy has on fission and n-capture cross-section is ignored. Furthermore, the effect of in-growth of neutron-absorbing nuclei ("poisons") is ignored. This exercise is not intended for - and will not convincingly model - the neutron inventory of a nuclear reactor. These limitations should be made clear for the students. This said, the simulation is quite useful in explaining and illustrating all the problems associated with nuclear waste.

## Provided Material

Below you will find teaching material intended to be handed out to the students and guides for teachers on how to set up and execute the exercise. The student material can either be used as is or incorporated into your own learning material (e.g. a book of exercises or an student management platform). Feel free to use this material as you would like, provided it is for non-commercial purposes. If you develop the material further, we would much appreciate it if you could

share the additional material here on NucWik. Likewise would we be very interested in your experience from using this exercise and material - please share!

## Guides and Material for Students

- [General introduction and student guide for this exercise](#). This is the hand-out for your students they should read as preparation for the exercise. In addition, you might want to point out relevant parts of your course book which they should read or already know before performing this exercise. It's written as a NucWik page and can be printed out directly or copied into your own delivery format.
- **Extra help for the students:** Depending on how much time you plan your students to spend on this exercise, you can provide some further help by providing them with a readymade input file (containing relevant nuclear data) and examples of how the data can be organized in the software. The best learning outcome is probably achieved if they figure out the data structure and assemble the data themselves, but it will probably take them many hours to do so. These tips were assembled by student Johannes Ø. Matsdal at [UiO](#) when he made a pilot for this exercise.
  - Here is an example of how the relevant parameters for the nuclei in the actinide region could be assembled: [ [Excel file](#) | [Text file](#) ]
  - Some programming tips, based on using the LabView language from [National Instruments](#): [ [Tips for using the input file](#) | [Tips for how to organize the data in LabView](#) ]

## Guides and Material for Teachers


- [Teachers guide for this exercise](#)
- [General description of Computing in Science Education \(CSE\) exercises](#) and how to use them in teaching (also linked at the top of this page).
- **Ready to go programs and source code** for performing this exercise:
  - LabView program written by Johannes Ø. Matsdal at [UiO](#): [ [Software Code](#) ] [ [Executable](#) ] [ [Explanations](#) ]. Notice that you either need the LabView plug-in (2014 SP1 version) or LabView system installed to run the program, it is explained in the "Explanations" link.
- [Examples of Obtained Results](#), from Johannes Ø. Matsdal (who wrote the pilot program to test out the concept)
- [Compilation of experiences and advice from teachers and students](#). Here we assemble feed-back from teachers and students that have performed the exercise. Please help us to make this a long and useful list!

## Development History and Contact Person


This exercise was conceived by Jon Petter Omtvedt ([UiO](#)) as part of the [CINCH project](#). The initial development was made by Hans V. Lerum ([UiO](#)) and Omtvedt in 2014/2015. Pilot programs were written by Johannes Ø. Matsdal and Håkon Becstrøm during the summer of 2015, they also added many refinements and added ideas.

If you have comments, suggestions, examples of programs (in any language) or anything else relevant, please write in the comment page (use the NucWik page commenting tool, but you must be a registered user) or send your feed-back to [NucWik-post@kjemi.uio.no](mailto:NucWik-post@kjemi.uio.no). The teaching



material for this particular CSE exercise is managed and updated by: [Jon Petter Omtvedt](#) . Feel free to contact me directly if you want to discuss this exercise or need help implementing it.

 Add Discussion

Contributions to <https://nucwik.wikispaces.com/> are licensed under a [Creative Commons Attribution Share-Alike 3.0 License](#). 

Portions not contributed by visitors are Copyright 2015 Tangient LLC

# Student Guide - Introduction to Simulation of Transuranium Waste Generation

The purpose of this exercise is to study how the nuclear waste that is not fission fragments is generated in a nuclear reactor. For this purpose a computer program will be written that simulates the nuclear reactions and disintegrations inside a nuclear-reactor core loaded with a specific fuel mixture. To keep the exercise doable within the allotted time we will introduce a number of simplifications. Nevertheless the simulation will provide important understanding and insight into the dynamics of a running nuclear reactor. The main simplification used is that only thermal neutrons are present in our simulated reactor.

## Learning Outcome

Performing this exercise should provide you with:

- A good understanding of the dynamics inside a reactor core (how the decay and in-growth of all the different nuclei is interlinked and depend on each other).
- A fundamental understanding of how the neutron flux will change the waste composition.
- A good understanding of how the waste composition changes as a function of time (both during irradiation and after).
- A good understanding of how the fuel composition will dictate the waste composition.
- Develop a more intimate feel for the basic workings of nuclear reactor.
- In addition, you will learn how complex technology can be investigated by computer simulations.

## Limitations

Many aspects of constructing and operating a nuclear reactor is not considered in this exercise. In particular, the importance of considering the effect neutron energy has on fission and n-capture cross-section is ignored. Furthermore, the effect of in-growth of neutron-absorbing nuclei ("poisons") is ignored. This exercise is not intended for - and will not convincingly model - the neutron inventory of a nuclear reactor. This said, the simulation is quite useful in explaining and illustrating all the problems associated with nuclear waste.

## Introduction

In a nuclear reactor one should think that only fission fragments should be left after having "burned" the nuclear fuel. However, this is not true since neutron capture in the fuel also will produce heavier actinides. In fact, transurans produced when burning uranium fuel is one of the major obstacles in handling spent nuclear fuel. An additional concern to the radiotoxicity of the waste is also the accumulation of plutonium that can be used for nuclear weapons.

To do laboratory tests with fuel mixtures to investigate the composition of the nuclear waste is extremely difficult and expensive, for obvious reasons. Therefore, we can not do such investigations as hands-on exercises in a course, but what we can do is to use a computer to simulate the processes that takes place inside the reactor core. To do so at an accuracy level that exactly models the reactor is highly complex and computer intensive, but if we accept a slightly less

accuracy level the simulations can be made simple enough to be run on an ordinary computer within a reasonable time. This is what we will do in this exercise.

## Theory

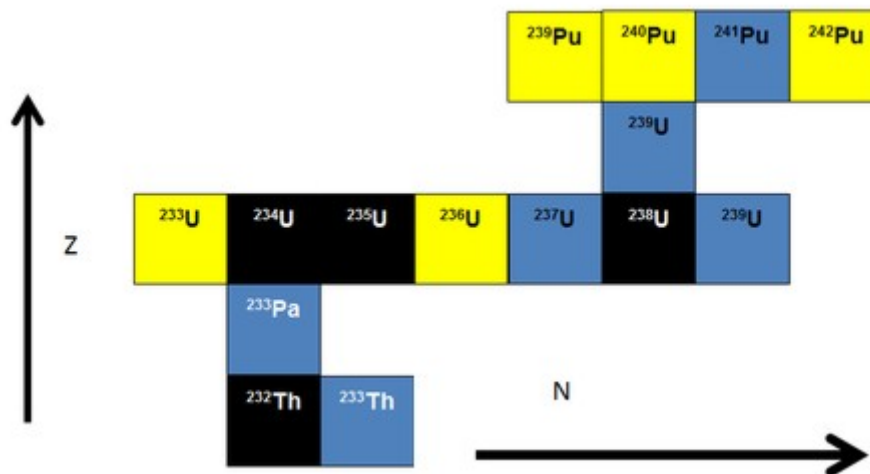
You might want to read up on the following theory before you start this exercise:

- Decay laws and mother-daughter relationships
- Nuclear reactions and cross-sections
- Fission and nuclear reactors

You should be able to find textbook material on these topics in your course book(s), on the internet and also on NucWik. In the following we assume you have a good basic knowledge of these topics.

As the fission process proceeds inside the reactor core, both fission fragments (the two halves left over after splitting the fissionable nuclei) and high-energy neutrons are released. The neutrons are, in an ordinary reactor, "cooled down" (i.e. their energy is reduced) until they only have "thermal" velocities (i.e. move with the same speed as any other atom at that particular temperature). The reason the neutron must be cooled down is because they are more reactive (have a higher cross-section) as thermal neutrons than as high-energy neutrons. Of course, in a real reactor also the high-energy neutrons will to some extent be reactive, but in this exercise we will ignore that - we only consider thermal neutrons.

The thermal neutrons will induce fission in the fuel nuclei and thus sustain a chain-reaction that keeps the reactor running.



However, thermal neutrons will also be captured by nuclei present in the fuel. This can be the fissionable nuclei (like  $^{235}\text{U}$ ) or nuclei just being present in the fuel (like  $^{238}\text{U}$ ). This process will lead to a buildup of a large number of heavy nuclei (not fission fragments) that were not present in the fuel at the start. For example if  $^{238}\text{U}$  captures a neutron  $^{239}\text{U}$  is produced. Now, these new nuclei are radioactive and will disintegrate according to their respective half-lives. This leads to creation of even more new nuclei. For instance, the  $^{239}\text{U}$  just mentioned will disintegrate by beta-minus decay and create  $^{239}\text{Np}$ . In principle, the  $^{239}\text{U}$  - if it survives long enough before disintegrating - could also capture a second neutron and that would produce  $^{240}\text{U}$ . If the reactor is operated long enough, which is normally the case, the buildup of all these transuranium nuclei can be quite extensive. Since some of them have very long half-lives they add significantly to the time the nuclear waste is going to be radiotoxic. A simplified picture of the most important nuclei being produced in a nuclear reactor burning uranium or thorium is shown to the right. The black nuclei might be present in the fuel at the start (we will investigate different mixtures), the yellow ones decay by alpha emission and the blue ones with beta-minus emission. You should check out your nuclear chart to see more details and try to understand the dynamics.

## Simulation Algorithm

Instead of all the coupled equations linking everything together, we will take the following approach:

1. We set up a matrix representing the relevant part of the nuclear chart (e.g. from thorium and up). Each element in the matrix represent one nuclei (like the nuclear chart).
2. We will assume a constant neutron flux during the entire operation (gross simplification), and only thermal neutrons at that. We assume the reactor remains critical until we switch it of.
3. Each matrix element (representing a given nuclei) should specify the amount of that nuclei being present in our "reactor". In addition, you would probably want to store relevant parameters for each nuclei in the same matrix element (using a suitable feature of your programming language to do so (e.g. in LabView we will use a "cluster"). The relevant parameters are:
  1. The cross-section for neutron capture
  2. The cross-section for neutron induced fission
  3. The half-life of the nuclei
  4. The type of decay, including branching information if there are more than one type of decay.
  5. In addition, you might find it useful to also write down the atomic mass number (can also be calculated from position in matrix)
  6. Depending on how you have defined your matrix, the N and Z for the nucleus can be useful
4. Initially, only the nuclei in the fuel will be present, all the other nuclei (matrix elements) will be empty (zero).
5. We will now use the computer to "update" the matrix as the reactor is running. We will do this with a fixed time interval between each update. Each time the computer "updates" it should do the following for each single matrix element:
  1. Use the given neutron flux and calculate how much has fissioned in the last time interval (if fission is possible). Subtract this amount.
  2. Use the given neutron flux and calculate how much of the neighbouring nucleus has been produced due to n-capture. Update both the current matrix element and the neighbouring one accordingly.
  3. Calculate how much of the nuclei has decayed during the last time interval. Subtract this amount and add it to correct matrix element according to the type of decay.
6. Repeatedly update the matrix until you have reached the desired operation time.

Provided the time intervals are kept short (experiment!) we should not get any big error due to changes taking place within a time interval. E.g. if the time interval is longer than the half life for a given nucleus, then that nucleus will be gone before the calculations are performed. However, in reality some of the nuclei could have undergone n-capture or fission before they decay. But if we keep the time interval short enough such effects should not be a problem.

You should also do the matrix update on a copy of the main matrix, because due to n-capture and decay some elements will change before the algorithm have updated them. When all elements have been updated you copy the data back into the main matrix.

You can solve this problem in most computer languages. Use the one that are available to you and that you have learned. We will not teach you programming skills in this exercise - we want you to

use skills acquired in previous courses. Remember to test each part of your code thoroughly and advance in small steps, making sure that each step is working correctly before you proceed.

## Problems to Investigate

Once your code work correctly, you can start to investigate how a nuclear reactor behaves under different conditions. Investigate the following scenarios:

1. Effects of fuel composition. For a thermal n-flux of  $10^{15}$  n/s and one year reactor operation, investigate the following fuels
  1. 3% enriched  $^{235}\text{U}$
  2. Pure  $^{235}\text{U}$
  3. 20%  $^{233}\text{U}$  and 80%  $^{232}\text{Th}$
  4. Pure  $^{239}\text{Pu}$
2. Investigate the composition of the waste for the different fuels after 1, 10, 100, 1,000 and 100,000 years storage. You do this by "running" the reactor with zero n-flux. For the long times you might want to increase the time-step size (why?).
3. For the 3% enriched  $^{235}\text{U}$  fuel, investigate the effect of different n-fluxes: Calculate waste after one year operation for  $10^{14}$ ,  $10^{16}$  and  $10^{18}$  n/s.

The results should be represented graphically and you should write a report which discuss the results and compare them to each other. What have you learned about operation of reactors and the resulting waste composition?

## Some Further Programming Tips and Suggestions

### Your program should be able to do:

1. The n-flux used in the program must be changeable
2. The program will be much more easy to use (and debug!) if it can fetch initial data and parameters from a file. The file should be written in ascii (plain text) so you can edit it with a normal text editor (take care that you store it as plain text if you use an advanced text editor like MS Word).
3. Similarly, it will be convenient if the program outputs the final results in a text file which can be inspected, printed and imported into suitable plotting software.

### Added features could be:

1. Stop the calculations before the total time has elapsed (in case it takes for ever to do the calculations)
2. Have the ability to display the result in different units: Bq or GBq, kg or g and number of atoms
3. Graphical display of trends

## Additional Exercises

1. Compare the waste from the previous reactor to a reactor using: 20% Thorium 4%  $^{235}\text{U}$  and 76%  $^{238}\text{U}$  as fuel. Show the activity of Th, Pu, Am and Cm after: 1 year, 10 years, 100 years and 1000 years.

2. You have one tonne MOX fuel with 7% plutonium and 93% Uranium. How much will the waste composition change with the flux of:  $10^{10}$ ,  $10^{12}$  and  $10^{14}$ ? The composition of the plutonium and uranium is given in the tables below.


The plutonium used consist of

Nuclei name	Amount wt%
238-Pu	1
239-Pu	60
240-Pu	25
241-Pu	9
242-Pu	5

And the uranium consist of

Nuclei name	Amount wt%
234-U	0,003
235-U	0,25
236-U	0,001
238-U	99,74

 Add Discussion

Contributions to <https://nucwik.wikispaces.com/> are licensed under a [Creative Commons Attribution Share-Alike 3.0 License](#). 

Portions not contributed by visitors are Copyright 2015 Tangient LLC

# CSE Teachers Guide - Simulation of Transuranium Waste Generation

Understanding the complexities involved in a nuclear reactor is demanding. As in most other such cases when teaching complex concepts, enabling the students to actually work the problem (or, as in this case, a part of it) will help them to grasp the problems and underlying concepts more quickly and probably better.

## What are the Teaching Benefits from using this CSE exercise?

The exercise presented here uses the concept of Computing in Science Education (CSE) to teach the students about transuranium waste accumulation in a nuclear (power) reactor. They will need to use their computing skills (which we assume they have) to write a program that simulates the nuclear reactions inside the reactor core (hence learning about fission, nuclear decay paths, and n-capture and the importance of cross-sections in steering these processes). If such a complex process is described using normal differential equations, the resulting set of fantastically complex coupled equations is far too demanding to handle for most Nuclear and Radiochemistry (NRC) students (and perhaps also their teachers..), nor would it be time inside normal curricula to handle such complex mathematics. Traditionally the teacher only explain how difficult it would be, explain how transuranium waste is building up and adding difficulty in the waste handling process and then move on to the next subject. The approach taken for this exercise is to use the number crunching capabilities of the modern desktop computer to dispense with the complex coupled equations and instead just calculate the change in the amount of the relevant nuclei in many small time-steps (e.g. each minute during a whole year of operation). Definitely not something you would like to do by hand, but the computer doesn't mind!

With this exercise the students will be activated and work out for themselves solutions that give them insights into the following:

- Fission, nuclear decay paths, and n-capture and the importance of cross-sections in steering these processes (as mentioned above)
- Where to find relevant nuclear parameters like cross-section and decay branching. To keep it manageable and the programming easy we have simplified the reactor to only have thermal neutrons (cross sections can be fetched from the nuclear chart and no neutron-energy complexity have to be considered).
- 

Once the program is up and running, the students can use it to investigate:

- The dynamics of transuranium waste build-up in a nuclear reactor. (Of course, the exercise can be extended to also consider the fission fragment waste, but that will make the task much bigger).
- Study the effects fuel composition has on the production of waste (in particular, the effect of having  $^{238}\text{U}$  present is easy to study with this simulation).
- By turning off the reactor (zero neutron flux) the simulation can be run when only decay occurs (fission and delayed neutrons are ignored), thus studying the dynamics the composition of nuclear waste as as function of timer.
- Study the effect of neutron flux magnitude on the composition and changes in the fuel (and waste).



## Time needed

How much time the students will need highly depend on the organization and level of guidance. Certainly it will also depend on the programming skills of the students. At University of Oslo we ran a pilot project where a bachelor student (Johannes Ø. Matsdal) was engaged to develop this exercise and write a working program. He used about 3 weeks to make the program work. His background was from a 10 cp course in Radioactivity and Radiochemistry (UiO course code KJM3900). With such an approach, where part of the effort headed off in the wrong direction since ideas and experiences was gained along the way, a large chunk of student time will be consumed by this project. This can seriously be reduced by providing a step-by-step guide that ensures that the student always head in the right direction. The problem can also be split into several separate steps that could each constitute a stand-alone exercise, but exercises that eventually would add up to provide the waste generation simulation program.

This kind of CSE exercise would probably also benefit from letting the students work in teams, where they can use their collective programming skills and work out problems and approaches by discussions among themselves. Furthermore, that would enable the computer geeks (there will nearly always be some) to work with others not so familiar with computer programming. The team should then divide tasks between themselves enabling everybody, based on their capabilities, to contribute (programming, putting together input data, testing the program, working examples, managing the group, plotting results, writing reports, etc., etc.).

Finally, if a readymade program (a little contrary to the CSE concept, but whatever works..) or maybe better part of the code is provided, the needed time can be seriously reduced. You, as the teacher, will have to work out an implementation that is right for your course and situation (and we would love to hear about your experience with it!).

Hopefully, as teachers use this concept for teaching about nuclear reactors and waste, we can provide examples of actual implementations here.

This is a rather large project and time the student should needs, will be 2.5-4 weeks (not including the weekends), depending on their background in programming and the language used.

## Choice of Programming Language

Ideally both you - the teacher - and the students should be proficient in the chosen programming language. If you do not have enough programming experience yourself, you can engage an assistant teacher (student?) to help out with the programming. If your students have extensive programming skills, you can depend on them to be able to solve the computer part of the problem. But this might be a little risky and probably demand a rather brave teacher..

There is no reason why not most normal programming languages should be applicable to this exercise (as long as they can handle mathematics properly). Watch out for the precision used in handling real numbers, round-off errors can give very strange behavior which can be difficult to identify unless you look out for it. Your students are unlikely to have experience with this.

This exercise is probably to complex to be suitable as a task for learning computer programming. The students writing the program should have all the basic understanding of writing computer code and how to construct the basic algorithms (i.e. know about terms like: For-loops, If-else constructs, while-loops, arrays, etc. It would probably be better to use object-oriented programming methods,

so such knowledge would be an advantage.

## Data Presentation and Reports

For certain, this exercise should not just be about making the program work (students tend to put much focus on this and forget the real purpose of writing the program in the first place..). Thus, it is important that once the program is up and running, it will be used to work real problems related to nuclear waste generation. These problems should be related to expected learning outcomes for the course. It is also a good opportunity to train the students in presenting experimental data properly and write proper scientific reports. Make sure the students have good tools for producing proper scientific graphs.

## Ideas for a step-by-step implementation

Below is a layout of a step-by-step approach to implementing this CSE exercise. It would probably be a good idea to set deadlines for each step and have some sort of delivery to mark completion of each step.

### Step 1: Define the problem

The student or student group should ensure that they understand the problem they are going to solve. Read up on relevant theory, select programming language and start thinking about how to approach the writing of the program and organizing the data. If in a group, how to divide tasks and make a project management plan.

### Step 2: Part one of the Computer Program - Organizing the Data

Decide on how the data, which has been defined in Step 1, should be stored inside the program. A good way to do this is by using a two-dimensional matrix structure like the nuclear chart where each element is an "object" or "cluster" containing all relevant data for the nucleus it represents, including the amount present in the fuel/waste. The position inside the matrix is determined by the nucleus' N and Z numbers). Make subroutines that:

- initialize the matrix,
- allow you to inspect any given element (nucleus) in the matrix
- allow you to edit any given element (nucleus parameters) in the matrix
- load data into the matrix from an input file
- write data from the matrix back into such an input file

Optionally the following would also be helpful:

- A matrix view showing the amount of each nucleus
- Alternatively (or in addition) a histogram type of view would be a nice touch

In addition, subroutines for the following tasks will be useful:

- Calculate N and Z for a given nucleus (specify mass-number A and element symbol as input)
- Generate nucleus name (A and element symbol) for given N and Z
- Calculate matrix position for given N and Z
- Calculate N and Z for a given matrix position

### Step 3: Part two of the Computer Program - Nuclear Reactions

Make a subroutine "CalcNucleusChange" which for a specified time interval calculate for a given matrix element (nucleus) how much of it will undergo:

- fission
- n-capture
- radioactive decay (take branching into account)

The output from this subroutine should be directed into a copy of the matrix where not only the changes for the given nucleus is reflected but also all the other nuclei (matrix elements) which are being populated through n-capture and radioactive decay. The subroutine need to be thoroughly tested and its correct working verified. Any error in this subroutine will make the program flawed. Watch out for decays which produce nuclei outside the defined matrix (unless you have chosen to represent the whole nuclear chart) and also nuclei that are unknown (unknown half-life).

Make a subroutine that for a given time interval traverse the complete matrix and call the "CalcNucleusChange" subroutine for each element to calculate the complete change in the matrix during this time interval. As input for CalcNucleusChange the original matrix should be used, all output should be directed into a single copy of the original matrix. Make sure that you copy the data from the original matrix into the new one before you start traversing the matrix, otherwise it will only contain the changes and not the total amounts.

### Step 4: Finished Computer Program

Make a loop that will call on the subroutine that updates the matrix for one time-step as many times it will take to add up to the time you would like the reactor to "run". For short times and many steps this can take quite some time. Ensure that the users are kept updated on progress and that the computer did not "die". It would be good to provide the user with a cancel button to halt calculations that takes too long time.

The program should keep track of "fuel history", i.e. a record of "reactor" operation time and, if asked for "cool down time" (operation with zero neutron flux) to ensure that simulations are properly recorded.

This step should include a thorough test of the complete program to ensure that it is ready for the simulations and to avoid that one at a later stage will need to go back and do "bug-fixes".

### Step 5: Performing the Simulations

Now the students should use the program to simulate the operation of different types of nuclear reactors and analyze the consequences the initial fuel mixture will have on the generated waste. This has been described in detail elsewhere and will not be repeated here.

### Step 6: Presenting Data and Reporting

The amount of data produced in Step 5 makes a good opportunity to train the students in how to present their data (graphs and histograms, tables if appropriate, etc.) and writing up a report with their findings and conclusions.

This step could (should) also include a discussion about the limitations of the approach taken in this exercise. This of course is first and foremost the simplification that only thermal neutrons are considered and that the neutron flux is assumed constant throughout the reactor core and as a

function of time.

## Concluding remarks and further ideas

We would much appreciate that you share your experience - good or bad - with this (and other) exercises on NucWik. The basic idea about NucWik is that we will do better teaching if we work together and share, please join us in this effort. A practical way to do this would be if you ask your students, as part of their final report, to also include their thoughts about this CSE exercise. Your own comments together with this will be highly appreciated!

At this time we would also like to point out that wikispaces, on which NucWik operates, is a wiki tool-provider dedicated for teaching. It is perfectly possible to use either your own wiki set up on the wikispaces site or just NucWik itself as a platform for collaboration and reporting for your students. In this way, especially if you use NucWik, all the results would be available to other teachers (and students). Please contact us at [NucWik-post@kjemi.uio.no](mailto:NucWik-post@kjemi.uio.no) if you have questions or need help setting up such a scheme.


Quick links: [ [Introduction to this Exercise](#) | [Introduction to CSE](#) | [Student Guide](#) ]

## Development History and Contact Person

This exercise was conceived by Jon Petter Omtvedt ([UiO](#)) as part of the [CINCH project](#). The initial development was made by Hans V. Lerum ([UiO](#)) and Omtvedt in 2014/2015. Pilot programs were written by Johannes Ø. Matsdal and Håkon Becstrøm during the summer of 2015, they also added many refinements and ideas.

If you have comments, suggestions, examples of programs (in any language) or anything else relevant, please write in the comment page (use the NucWik page commenting tool, but you must be a registered user) or send your feed-back to [NucWik-post@kjemi.uio.no](mailto:NucWik-post@kjemi.uio.no). The teaching material for this particular CSE exercise is managed and updated by: [Jon Petter Omtvedt](#). Feel free to contact me directly if you want to discuss this exercise or need help implementing it.

 Add Discussion

Contributions to <https://nucwik.wikispaces.com/> are licensed under a [Creative Commons Attribution Share-Alike 3.0 License](#). 

Portions not contributed by visitors are Copyright 2015 Tangient LLC

# CSE Exercise - Simulation of Radionuclide Genetic-Dependency (Mother-Daughter relationship)

In this Computing in Science Education (CSE) exercise we mainly use the computer's number crunching capability to quickly plot the total disintegration rate from genetically dependent mother-daughter nuclei. To be more realistic we will also optionally add a constant "background" to the plots. In this way the plot will be comparable to what you actually would observe from a detector measuring observing radiative events caused by disintegrations from either the mother nucleus, the daughter nucleus or sporadic events from the general background radiation.



There are two principal routes to calculate the disintegration rate from the mother and daughter nuclei: Either you use the analytic solutions to the equations, or you can solve the differential equation representing the mother-daughter genetic dependence. The first one is the easiest and most practical; the second will train you in numerical methods for solving differential equations. In both cases you will be able to graphically investigate the dynamics in mother-daughter relationship and the principles behind a radionuclide generator if you add plotting capabilities to your program.


Follow this link for a general description of [Computers in Education \(CiE\) exercises](#).

## Learning Outcome

In addition to getting more experienced in solving differential equations numerically, the student will get the following learning outcomes in NRC (Nuclear and Radiochemistry) from this exercise:

- Practical experience in handling decay equations, providing a more in-depth understanding of what they represent.
- Improved understanding of genetically dependent nuclei, in particular for transient and secular equilibrium and the conditions that will lead to such equilibria.
- A good understanding of the dynamics of a radionuclide generator.

## Guides and material for students

- [Student guide to this exercise](#)
- **Theory and Background**
  - The numerical solution of differential equations will not be explained here but you (the teacher) should point your students to the right course material. A good introduction to one basic (but good) method - Euler's method - is available from Wikipedia (as per October 2015): [https://en.wikipedia.org/wiki/Euler\\_method](https://en.wikipedia.org/wiki/Euler_method) . Euler's method is suggested used in the [Student Guide](#) above, together with the analytical solution.
  - You (the teacher) should point out relevant pages in your course book to your students. In some much used books this would be:
    - Loveland et al. "Modern Nuclear Chemistry": Chap. 3: Radioactive Decay Kinetics

- Choppin et al: "Radiochemistry and Nuclear Chemistry" (fourth edition): Chap 5: Unstable and Radioactive Decay, sub-chapters 5.11-5.16.
- Lieser: "Nuclear and Radiochemistry: Fundamentals and Applications" (second edition): Chapter 4. Radioactive Decay
- NucWik pages with relevant theory explanations:
  - [Disintegration](#)
  - [Principle Behind Mother-Daughter Relationship](#)

## Guides and material for teachers


- Once the student has the program up and running, it should be rather straight forward to walk him/her through relevant examples:
  - Short-lived mother, long-lived daughter (no equilibrium)
  - Short-lived mother, short-lived daughter (no equilibrium)
  - Long-lived mother, short-lived daughter (transient equilibrium,  $T_{1/2}(\text{mother}) \gg T_{1/2}(\text{daughter})$ )
  - Long-lived mother, short-lived daughter (secular equilibrium,  $T_{1/2}(\text{mother}) > 10^4 \times T_{1/2}(\text{daughter})$ )
  - Examples of operation of a radionuclide generator, e.g. the  $^{99}\text{Mo} \rightarrow ^{99\text{m}}\text{Tc}$  generator. How much activity will you get after one, three and 10 daughter half-lives? What happens if you milk the generator two times in a row?
- Readymade source code and executables (intended to enable teachers considering to use this exercise to evaluate it more easily):
  - LabView example: [ [Source code \(VI\)](#) ] [ [Executable](#) ] [ [Wiring Diagram](#) ] [ [Running VI Screen Shot](#) ]
- Tips for how this CSE exercise or parts of it could be used in your teaching:
  - If you already have the program, or if you write it yourself, you can allow the students to play with it to learn about mother-daughter relationships even if they are not able or do not have time to write the program themselves. But then it would not be CSE, though...
  - Alternatively, you can use the analytically derived solution in spreadsheet type programs like MicroSoft Excel or even (much) better data handling programs like [Origin from OriginLab](#)
  - This CSE exercise (or the alternatives mentioned above) can be used as preparation for laboratory exercises where mother-daughter relationships are being used or investigated. Examples are:
    - Laboratory exercises with  $^{99\text{m}}\text{Tc}$  generators: [ [Liquid-liquid Extraction and Measurement of  \$^{99\text{m}}\text{Tc}\$](#)  ]
    - Laboratory exercises with uranium daughters: [  [\$^{234\text{m}}\text{Pa}\$  Radionuclide Generator \(Method 1\)](#) ] [  [\$^{234\text{m}}\text{Pa}\$  Radionuclide Generator \(Method 2\)](#) ]
    - Other relevant laboratory exercises: [ [Determination of the  \$^{238}\text{U}/^{234}\text{U}\$  Ratio](#) ]
    - You should also search NucWik for Laboratory exercises added after this was last updated (searching the tags is the best way, provided those that entered the exercise added the relevant tags to their pages).
- [Compilation of feedback from teachers and students](#). Here we assemble feed-back from teachers and students that have performed the exercise. Please help us to make this a long and useful list!

## Development History and Contact Person

This exercise was conceived by Jon Petter Omtvedt ([UiO](#)) and used for teaching in Nuclear Chemistry at University of Oslo since many years. It was included here at NucWik as part of the [CINCH project](#). Håkon Beckstrøm (UiO) added some comments and experiences from using it for teaching how to solve differential equations by numerical methods.

If you have comments, suggestions, examples of programs (in any language) or anything else relevant, please write in the comment page (use the NucWik page commenting tool, but you must be a registered user) or send your feed-back to [NucWik-post@kjemi.uio.no](mailto:NucWik-post@kjemi.uio.no). The teaching material for this particular CSE exercise is managed and updated by: [Jon Petter Omtvedt](#). Feel free to contact me directly if you want to discuss this exercise or need help implementing it.

 Add Discussion

Contributions to <https://nucwik.wikispaces.com/> are licensed under a [Creative Commons Attribution Share-Alike 3.0 License](#). 

Portions not contributed by visitors are Copyright 2015 Tangient LLC



# Student Guide - Simulation of Genetically Dependent Radionuclids (Mother-Daughter relationship) CSE Exercise

## Introduction

There are many cases where where a radionuclide decays into a second radionuclide or even long series of consecutive radionuclids. In such cases the disintegration rate of the second radionuclide (we usually refer to this as the "daughter") depend on the decay rate of the first radionuclide (the "mother"). You will find this explained in great detail in your course book and on-line (Google it!).



Number of mother nuclei is:  $N_1 = N_{1,0}e^{-\lambda_1 t}$

Disintegration rate of mother is:  $A_1 = \frac{dN_1}{dt} = \lambda_1 N_1$

In a time interval  $dt$  the increase  $dN_2$  in daughter is:

$dN_2 = (\lambda_1 N_1 - \lambda_2 N_2) dt$

In-growth rate of daughter

Decay rate of daughter

$\Rightarrow \frac{dN_2}{dt} + \lambda_2 N_2 - \lambda_1 N_1 = 0$

$N_1 = N_{1,0}e^{-\lambda_1 t}$

$\Rightarrow \frac{dN_2}{dt} + \lambda_2 N_2 - \lambda_1 N_{1,0}e^{-\lambda_1 t} = 0$

To the right is a brief review of how the differential equation for the rate-change expression for the amount of daughter is derived (click on the picture for a larger version). The parameters have the usual names,  $N$  for the amount (number) of each nuclei,  $\lambda$  is the disintegration constant, and the indexes is 1 for mother and 2 for daughter. The time,  $t$ , is the independent variable.

In this exercise you can either use numerical methods to solve this equation or you can use the analytically derived mathematical solution. We suggest that you do both.

## Analytical Solution

The analytical solution to the differential equation is:

$$A_2 = \frac{\lambda_2}{\lambda_2 - \lambda_1} A_1 (1 - e^{-(\lambda_2 - \lambda_1)t})$$

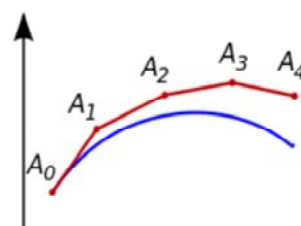
Here we have introduced the disintegration rate  $A=AN$  instead of the amounts of the two nuclei.

How this equation is derived is explained in basic math courses and has probably also been explained by your NRC teacher. It is not very complicated, but involves a few standard "tricks" and takes some time. We will not explain it here, but you can find it explained in e.g. Walter Lovelands's textbook "Modern Nuclear Chemistry" in Chapter 3.3 (page 67).

If you use this equation and write a program that plots the disintegration rate  $A_2$  of the daughter (you should also plot the rate  $A_2$  of the mother to get the complete picture), you will be able to learn a lot about radioactive equilibria by playing with the mother and daughter half-lives. This is explained below. In addition, you can train your skills in using numerical methods to solve differential equations. For this example, an analytical solution is possible, but for more complex problems the numerical methods might be the only solution. It will be a good exercise to use the numerical approach, since you can check the results against the solution obtained with the analytical formula.

## Numerical Solution - Euler's Method

There is several different ways to numerically solve a differential equation. One of them, and the one we suggest you use here, is Euler's method. This method is based on that we know initial value of the function we are seeking a solution for - which for us is the initial amount of the daughter nuclei - and that we have an expression for the derivative of this function. The derivative is of course our differential equation. Since the derivative is the slope of the unknown function we are seeking, we can estimate the value close to a known point (the initial point) by moving along the tangential line with the same slope. See the graphical



illustrations to the right (copied from [Wikipedia](#)). The blue line is the unknown function, the red line is the stepwise approximation starting in  $A_0$ .



The error in each step (the discrepancy between our approximated point and the true value) is going to increase the further away from the initial point we get. It is a first-order method, which means that the local error (error per step) is proportional to the square of the step size, and the overall error is proportional to the step size.

A good description of [Euler's method can be found on Wikipedia](#), please refer to this or to a course book in numerical methods for details.

For our problem, where we have a radioactive mother nucleus disintegrating into a radioactive daughter, we know the expression for the derivative  $\frac{\delta N(t)}{\delta t}$ , of the two curves representing the daughter and mother:

$$\frac{\delta N_1(t)}{\delta t} = -\lambda_1 N_1 \quad (1)$$

$$\frac{\delta N_2(t)}{\delta t} = \lambda_1 N_1 - \lambda_2 N_2 \quad (2)$$

We can then use Euler's method to calculate the amounts of the two nuclei as a function of time, assuming that the initial amount of the mother nucleus is  $N_{1,0}$  and that  $N_{2,0} = 0$  (i.e. we start with only mother present, like immediately after a chemical separation).

If we use Euler's method to approximate the amounts of mother and daughter with a step size of  $h$ , we can express the new value for the mother after any single step as:

$$N_1(t+h) \approx N_1(t) + h \left( \frac{\delta N_1(t)}{\delta t} \right) \quad (3)$$

$$N_1(t+h) \approx N_1(t) + h(-\lambda_1 N_1(t)) \quad (4)$$

The first step, when we start with  $t = t_0$ , would be:

$$N_1(t_0+h) \approx N_{1,0} + h(-\lambda_1 N_{1,0}) \quad (5)$$

Likewise for the daughter:

$$N_2(t+h) \approx N_2(t) + h \left( \frac{\delta N_2(t)}{\delta t} \right) \quad (6)$$

$$N_2(t+h) \approx N_2(t) + h(\lambda_1 N_1(t) - \lambda_2 N_2(t)) \quad (7)$$

The first step for the daughter would be:

$$N_2(t_0+h) \approx N_{2,0} + h(\lambda_1 N_{1,0} - \lambda_2 N_{2,0}) \quad (8)$$

which, since  $N_{2,0} = 0$ , is:

$$N_2(t_0+h) \approx h(\lambda_1 N_{1,0}) \quad (9)$$

Use equation (4) and (7) to calculate the amount of mother and in-growth of daughter you will get for a selected time span. Make sure that the step-size you use is small enough to make the stepwise error negligible.

### The Nuclear Chemistry Concepts you should Investigate

The numerical solution of the differential equation will enable you to plot the curves and illustrate the changes in disintegration rate of mother and daughter. In particular, you should look into the genetic scenarios suggested at the bottom of this page. Once you have code that enable you to calculate how much daughter you get as a function of time, you should use this to investigate graphically the genetic dependence of mother-daughter relationships with respect of mother and daughter half-life scenarios.

From the analytical solution to the mathematical problem you can see how the half-lives of the two nuclei determine the genetic relationship. In particular you should notice and investigate how the fraction with the  $\lambda$ s in front of  $A_1$  behaves for different combinations of mother and daughter half-lives. Likewise it is important to understand the role of the expression within the parenthesis - the "saturation factor". Work out the limit when  $t$  approaches infinite!

As a concrete and also very important example, you should in particular investigate the  $^{99}\text{Mo}/^{99\text{m}}\text{Tc}$ -radionuclide generator.  $^{99\text{m}}\text{Tc}$  is the most commonly used medical radionuclide in the world and the hospitals use generators with  $^{99}\text{Mo}$  attached to a column to produce the  $^{99\text{m}}\text{Tc}$ .

We suggest that you write your program step by step and thoroughly check that each step works before you continue. You might want to follow the following path:

**Step 1:** Initialize parameters and get input parameters (amount and half-life of mother, half-life of daughter). You might also want to allow the user to specify the time interval to be modelled (keep careful track of the time units you use so you do not mix them up).

**Step 2:** Use both the analytically derived equation (approach A) and the numerical solution (approach B) to calculate the amount of mother and daughter (activity  $A_1$  and  $A_2$ , respectively) as a function of time. Store the results in two arrays 1000 elements long. You might find it convenient to also have a third array where you store the time for the respective elements in the two first arrays.

**Step 3:** Now you should plot your data. If you use a programming language that do not have good support for plotting, you might instead opt for generating a text file with the numbers to be plotted and import them into some appropriate plotting software. The plots should be semi-logarithmic (i.e. logarithmic scale along the y-axis, but not along the x-axis). Why do we want this? Your plots should have three different curves: 1) Sum of  $A_1$ ,  $A_2$ , and background; 2) only  $A_1$ ; and 3) only  $A_2$ . Both x- and y-axes should be appropriately labeled.

**Step 4:** Add functionality to your program that allow you to either print or generate picture-files of your plots, as you will need to include them in your final report. If this is not possible with your choice of programming language you can do screen dumps, but this is never a good solution (nearly always too low-res).

**Extras:** Optionally you can select to also add a constant "background count rate", as a nuclear detector always will be exposed to this and therefore will be present and must be handled in a real measurement situation (you will most likely use the simulation program later when you do exercises that actually measure radioactivity).

#### Tips:

- The timestep for the numerical solution needs to be small enough for the method to be precise.

### Investigate Genetic Scenarios

Now you can start using your program to learn more about radioactive decay! Investigate the following scenarios where you have a daughter nucleus genetically dependent on the mother:

- Short-lived mother, long-lived daughter (no equilibrium)
- Short-lived mother, short-lived daughter (no equilibrium)
- Relatively long-lived mother, short-lived daughter (transient equilibrium,  $T_{1/2}(\text{mother}) \gg T_{1/2}(\text{daughter})$ )
- Very long-lived mother, short-lived daughter (secular equilibrium,  $T_{1/2}(\text{mother}) > 10^4 \times T_{1/2}(\text{daughter})$ )
- Examples of operation of a radionuclide generator, e.g. the  $^{99}\text{Mo} \rightarrow ^{99\text{m}}\text{Tc}$  generator. How much activity will you get after one, three and 10 daughter half-lives? What happens if you milk the generator two times in a row?

 Add Discussion

Contributions to <https://nucwik.wikispaces.com/> are licensed under a [Creative Commons Attribution Share-Alike 3.0 License](https://creativecommons.org/licenses/by-sa/3.0/).



Portions not contributed by visitors are Copyright 2015 Tangent LLC

# CSE Exercise - Simulating a Gamma-ray Detector

This CSE exercise will allow the student to write a program that simulates how a gamma detector works. The advantage over using a real detector is that in the simulation the different absorption mechanisms that determine detection of gamma rays can be turned on and off at will, something which of course not is possible in real life. This will provide the student with a better understanding of the detection process of gamma rays.



Follow this link for a general description of [Computing in Science Education \(CSE\) exercises](#).

## Learning Outcome

- Same as for using the readymade simulation program to study gamma-ray detection.
- Training in writing computer code to simulate a real system.

## Guides and Material for Students

- [Student guide on how to use the readymade Simulation Program](#). Here you have a description of the purpose of the program and how it can be used to study gamma-ray detection.
- [Student guide on how to write a program](#) that will simulate detection of gamma-rays (in LabView).

## Guides and Material for Teachers

- Source code for the gamma detector simulation program, written in LabView, is available here: [ [Source Code \(VI\)](#) ] [ [Executable](#) ]. Remember that to run the executable you will need the appropriate LabView run-time plug-in from National Instruments (version 2012 in this case).
- Tips and tricks - compilation of experience and advice from other teachers is collected here.

Add Discussion

Contributions to <https://nucwik.wikispaces.com/> are licensed under a [Creative Commons Attribution Share-Alike 3.0 License](#).

Portions not contributed by visitors are Copyright 2015 Tangient LLC

# Student Guide to a Computer Program which Simulate Gamma-ray Detection

Here an exercise is presented that uses a computer program to simulate how a detector absorbs and measures  $\gamma$  rays. Using the program enables you to investigate the absorption and detection processes in detail. It will help you to understand how a  $\gamma$  detector works and thus use it more efficiently.

## Background

Absorption of  $\gamma$  radiation occurs by three different effects: The photoelectric effect, the Compton effect and pair formation. A NaI crystal will emit light when  $\gamma$  radiation is absorbed. If we optically couple the crystal to a sensitive light-detector (a photo multiplier tube), we can measure the amount of light emitted and therefore also the  $\gamma$  radiation. In this way we can construct a  $\gamma$  detector and we call it a NaI-scintillator detector (to scintillate means emits light).

The amount of light is proportional to the amount of energy deposited in the detector. The light detector is so fast that it's able to measure the light from each individually absorbed  $\gamma$  ray. By analysing the pulse height from the detector (which is proportional to the amount of light) we can make a picture of the distribution of energy deposited in the detector. I.e. we make a histogram where each column represents the number of events within a small energy range - we will get a plot of event intensity as a function of energy deposited in the detector. Such a histogram we call a *spectrum*.

If we have a source which only emits  $\gamma$  rays with a single energy, its spectrum should look like Fig. 1. However, if we measure such a source with a NaI scintillation detector, the spectrum we get looks like Fig. 2. The reason is of course the manner  $\gamma$  rays are absorbed in the detector. A large fraction of the  $\gamma$ 's will be absorbed by the Compton effect and this severely distorts the spectrum. If the energy is high enough we will also get pair formation that further distorts the spectrum.

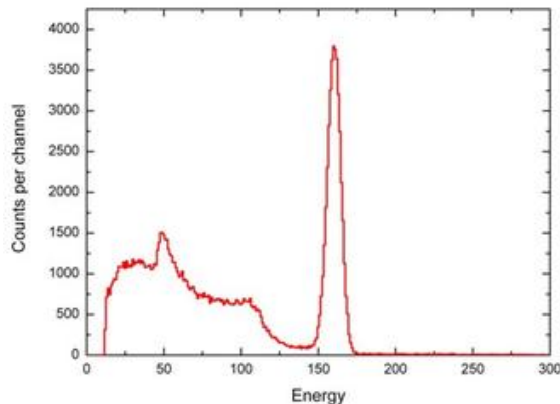


Fig. 1: Real spectrum measured with a NaI detector

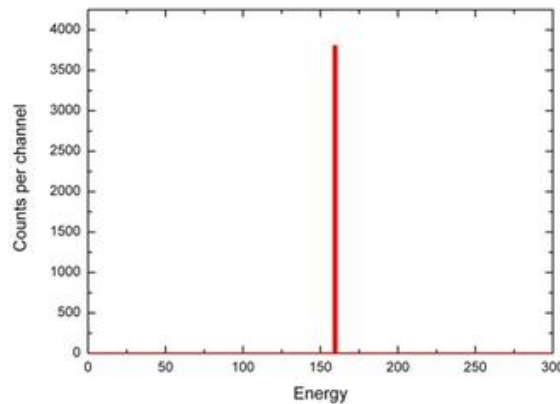
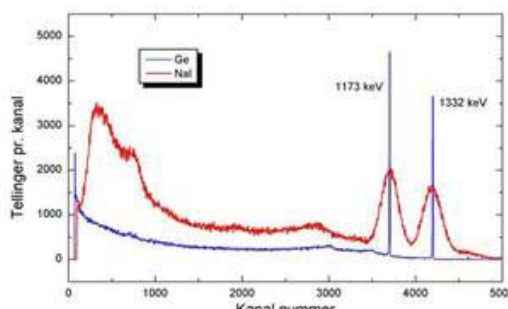


Fig. 2: Emission spectrum from mono-energetic  $\gamma$ -source

Other types of  $\gamma$  detectors work not by light emission (scintillation), but are based on semi-conductor technology. Here the  $\gamma$ -sensitive element is actually part of an electronic circuit and acts like an electrical diode. Radiation hitting the diode will generate a current pulse which is amplified and measured by the electronic circuit. Such detectors are usually named Ge-detectors since the Ge-detector crystal which constitutes the largest part of the diode is made of this element. Ge-detectors have much better energy resolution than NaI detectors, but are much more expensive.

For any kind of detector, the measurement of the generated signal (whether it is an electric signal or a light pulse) will never be 100% accurate. The error in the measurement will widen the peaks and generally smooth the spectrum. Thus, details will be more difficult to distinguish. The detector error is generally discussed in terms of the detectors *response function*. A Ge-detector has a better response function than a NaI detector. The simulation program presented here will enable you to



change the response function to see how it affects the measured spectrum. This is also something which generally can not be done with a real detector.

Fig. 3: Comparison between a NaI and a Ge detector

## Software

A program written in LabView can be downloaded. In order to run the program you must have the associated plug-in installed in your computer, the plug-in can be downloaded from National Instrument (you must register to do so, but it is free of charge). Once the plug-in is installed, you can run the simulation program.

Here are the relevant links:

- LabView Simulation Program (written by Jon Petter Omtvedt at UiO)
- National Instrument Plug-in
- Source code (not needed unless you want to change the program and/or add your own features). Please share any improvements you make with the NucWik community!

## Exercises

### Exercise 1: Detection of $\gamma$ radiation

In a real detector one can not turn the different absorption effects on and off - they will always be on. However, in this exercise we will use the simulation program and then you have full control of which effects should be active or not. You can even vary the intensity of the different effects.

The program or "virtual instrument" enables you to control four detector properties: The response function (responsible for the measurement error) and the absorption coefficients for the photo electrical effect, the Compton effect and pair formation. Furthermore the application let you chose if your source should emit  $\gamma$  rays with one, two or three different energies. And you can vary the intensity of which each energy is emitted with.

Perform the following experiments with the program, where you instruct it to use only a single  $\gamma$  energy with 662 keV of energy (that will correspond the the  $\gamma$  emission from a  $^{137}\text{Cs}$  source):

- Make a "perfect" spectrum, as it would be measured if only the photo electrical effect was present and no detector error. I.e. set the application's response function to 0, the photo electrical effect to 100%, the Compton effect to 0% and pair formation to 0%.
- Make a spectrum which has no detector error (response function = 0), but only absorbs  $\gamma$  rays with the Compton effect (photo electrical effect = 0%, Compton effect = 100% and pair formation = 0%). Compare this spectrum to the one you made in the previous step. Which one is most suitable for analysing the radiation energy and intensity from a  $\gamma$  source?
- Make a spectrum where only the photo electrical effect is at play, but select a response function of 100. Can you explain the shape of the response function? Tip: Some times the primary electron (the one which the  $\gamma$  ray transferred its energy to) will not excite the detector crystal as much as it should have done. This might be due to impurities in the crystal or that the electron exits the crystal before it has been stopped completely. In other words we will have some events which can have anything from 0% and up to 100% of the correct energy. For example, an electron which deposit 50% of its energy and then escapes the crystal will be registered as a count with only half of the "correct" energy.
- Make a spectrum with the response function set to 100, the photo electrical effect equal to 50% and the Compton effect equal to 50%. Compare this to a spectrum from a real NaI detector, as shown in Fig. 1 above.

For each experiment you should write down the parameters you have used and store the spectrum as an image file. In this way you can incorporate your spectra in your report (if your teachers demand you write one). You store the spectra by right clicking it and then choosing the menu command "Export Simplified Image". It's the same procedure as when you pick pictures from web-pages in most browsers.

### Exercise 2: Analysing spectra with multiple $\gamma$ lines

Use the simulation program you used in exercise 1, but this time you select three different  $\gamma$  ray energies: 400 keV (10% intensity), 450 keV (50%) and 800 keV (100%).

- Make a spectrum with an ideal detector (response function = 0, photo electrical effect = 100%, Compton effect = 0% and pair formation = 0%).
- Compare this spectrum with what you get if the response function is 10 and 100.
- What happens if you turn on the Compton effect?

### Conclusion


In your report's conclusion, you should try to explain what you have learned about  $\gamma$  spectroscopy. E.g. how does the detectors energy resolution (the response function) influence the quality/uncertainty of the spectrum? And what would be the advantage of selecting a detector made in such a way that the photo electrical effect is as dominant as possible?

### Development History and Contact Person

This exercise was originally conceived by Jon Petter Omtvedt ([UiO](#)) and used in radiochemistry courses at UiO. It was adapted for NucWik as part of the [CINCH project](#).

If you have comments, suggestions, similar programs (in any language) or anything else relevant, please write in the discussion box below (but you must be a registered user) or send your feed-back to [NucWik-post@kjemi.uio.no](mailto:NucWik-post@kjemi.uio.no).

 Add Discussion

Contributions to <https://nucwik.wikispaces.com/> are licensed under a [Creative Commons Attribution Share-Alike 3.0 License](#). 

Portions not contributed by visitors are Copyright 2015 Tangient LLC



# Student Guide on how to Write a Program for Simulation of Gamma Detection

In the [list of NucWik Simulation Exercises](#) there is an exercise that [simulate a gamma-ray detector](#), which allow you to investigate the details of how gamma rays are detected, i.e. the way gamma rays are absorbed in the detector crystal.



This exercise can be expanded into a [Computing in Science Education \(CSE\)](#) exercise where you write your own simulation program instead of using already written software. This will give you a more detailed feel for how the simulation works. In addition, you get training in how to use computers to simulate physical processes.

Here we will give you some tips on how to write such a simulation program. We assume you are familiar with absorption of gamma rays. If this is not the case, then please review it in an appropriate textbook.

There are many ways to simulate a gamma-ray detector. Here, we will try to write a program that when running actually will look like an acquisition system collecting data from a real gamma-ray detector (e.g. a scintillation-type detector like NaI or a semiconductor detector like a HPGe-detector). We will keep it rather simple, ignoring some physical effects, in order to not make this exercise too complex. You can always add sophistication to the program at a later stage.

We want a program that:

- Detects radiation from a source emitting gamma rays with one, two or three distinct gamma-energies. The relative intensity between the different gamma-rays can be specified at will (0-100%).
- The simulated detector will either absorb electromagnetic radiation through the photoelectric process (full energy absorption) or by the Compton process (an electron with part of the incident gamma-ray's energy and a new radiation quantum with the remaining energy generated). We will disregard pair-formation at this stage (i.e. the program will only work correctly for gamma-ray energies below 1022 keV). The relative ratio between the different processes can be user selected or turned off (by setting it to 0%).
- The simulation can be started and stopped like a real detector system, and after starting the data will slowly accumulate in a intensity (counts per channel) vs energy (channels or histogram bins sorted according to increasing energy) spectrum. The spectrum can also be cleared, exactly like in an ordinary acquisition system.
- The detectors response function, which determine how precisely it will measure the absorbed gamma-ray energy, can be varied to illustrate how this property will affect the spectrum.

Simplifications:

- The ratio between different absorption processes is in reality highly energy dependent. It would make the spectra more realistic, but we ignore it in this first version of the program.
- As already mentioned above, we ignore pair-formation.

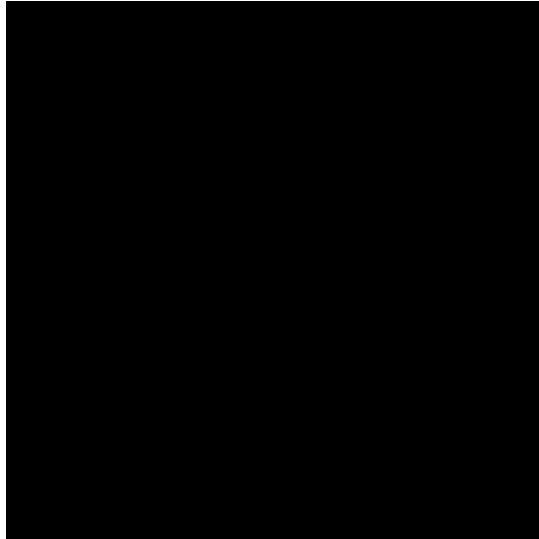


- The random emission of gamma rays from a disintegrating source is ignored. Instead we allow the program to run in a loop generating events as fast we can. If your computer is very powerful (and/or you write very efficient code) you may need to slow things down a bit to get a reasonable spectrum build-up.

To simulate the random process of nuclear disintegration we will use a random generator to generate detector "events" at random times and with random energies. However, we will filter these randomly generated events in such a way that their distribution will obey the rules of gamma-ray absorption in matter. This filtering we will have to do for each single event, otherwise the data acquisition will not look like in a real detector.

An easy way to generate a random event stream which, when a large number has been collected, will build up show a specific distribution pattern ("spectrum"), is to "roll a dice" for each event (with a given energy) where the event is either accepted or rejected depending on the likelihood of what will happen inside the detector when a gamma-ray with just that energy is absorbed.

At this stage you should understand that what a gamma-ray detector detects is the amount of the gamma-rays energy which has been transferred as kinetic energy to an lattice-electron in the sensitive part of the detector. This electron's kinetic energy is measured by the detection system and will show up as a single "event" in the spectrum, i.e. the spectrum channel representing that energy increases it's number of counts by one. The visual picture is a histogram with columns that slowly build in height. If you are not familiar with this we have a simple animation (to the right) that show how a spectrum is slowly being built up inside a "Multi Channel Analyzer (MCA)". (It's a MP4-format file, so your browser need to be able to play this format.)



## Writing the Program

One way to code this simulation is as follows:

Make a main loop (e.g. a while loop) which when the detector is "switched on" will generate events. Each generated event is processed as follows:

### Step 1: Determine which energy is emitted from the source

Determine which gamma (i.e. energy) should be generated, depending on the gamma's relative intensity. This step will always output one event. On a scale from 0 to 100 assign the three different gamma-energies a portion of the scale according to it's relative intensity. Get a random number between 0 and 100, and select the gamma that is represented within this part of the scale.

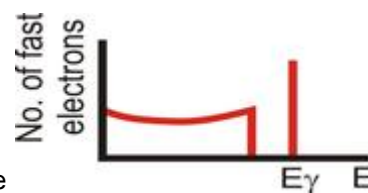
For example, if gamma-1 is emitted with a likelihood of 10%, gamma-2 with 50% and gamma-3 with 40%, The scale would be divided like this: 0-10 represent gamma-1, 11-60 represent gamma-2 and 61-100 represent gamma-3. Then, if the random number turns out to be 56,

gamma-2 will be selected. If it had been 76, it would have been gamma-3, etc.

### Step 2: Select absorption mechanism

The gamma ray emitted from the source (in Step 1), how will it be absorbed? Determine if it should be either the photoeffect or the Compton effect (as already explained, pair formation is ignored, but you could add it here at a later stage).

If the absorption is by the Compton effect you need to randomly select an angle between the electron and the Compton quantum, as any of these angles is possible. The energy measured by the detector system will be the electron energy generated by the selected angle. As you should know (or read up on the Compton effect), there is a specific equation that give you the relationship between the initial gamma-energy and the energy of the generated new quantum (frequently wrongly called a new gamma-ray, which it is not since it originates from outside the nucleus). Not all energies between zero and the original gamma-ray energy is possible for the ejected electron: there will be a maximum energy, which we refer to as the Compton edge. The distribution of electrons between zero energy and the Compton energy is not uniform, because each arc of angle cover different energy ranges. The distribution of Compton electrons will look like in the figure to the right, where we also have shown the foto-peak for comparison. You will get the right shape for the distribution by simply randomly selecting angles between 0 and 180-degrees.

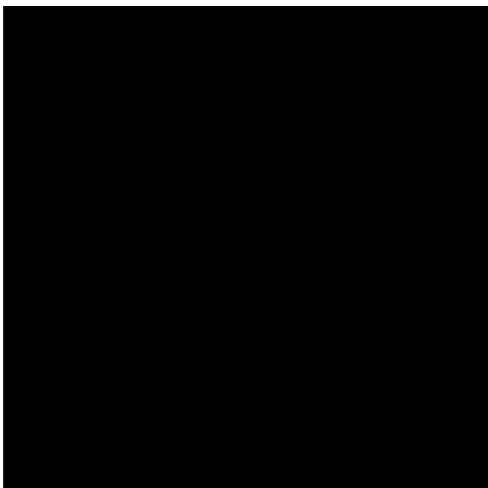


### Step 3: Apply the detector's response function

Apply the detectors response function, which will determine the width of the peaks in the spectrum. The response function is a gaussian like peak with a certain width (according the uncertainty the detector is able to measure with). In addition, it will have a small, but none-zero likelihood of generating a far too small response between anything from zero to the correct energy. This can be explained by wall effects or impurities in the detector crystall causing only a fraction of the kinetic energy on the electron actually produce a measureable signal. The shape of the function is shown to the right. You need to find an appropriate mathematical function to represent this.

For each loop (Step1 -> Step 2 -> Step 3) you increment the histogram that represent the spectrum with the new event. I.e. you increase the number of "counts" in the appropriate channel (histogram column) representing the given energy by 1 (see the video about the MCA spectrum above, if you have not already done so). Make sure you update the graph with the spectrum for each turn of the main loop, otherwise you will not get the impression of a live acquisition system.

That should be it! An example of a working program can be seen in the video below.




## Development History and Contact Person

This simulation was conceived by Jon Petter Omtvedt ([UiO](#)) and used for teaching in Nuclear Chemistry at University of Oslo since many years. It was included here at NucWik as part of the [CINCH project](#).

If you have comments, suggestions, examples of simulation software (in any computer language) or anything else relevant, please write in the comment page (use the NucWik page commenting tool, but you must be a registered user) or send your feed-back to [NucWik-post@kjemi.uio.no](mailto:NucWik-post@kjemi.uio.no).

The teaching material for this particular simulation exercise is managed and updated by: [Jon Petter Omtvedt](#). Feel free to contact me directly if you want to discuss this exercise or need help implementing it.

 Add Discussion

Contributions to <https://nucwik.wikispaces.com/> are licensed under a [Creative Commons Attribution Share-Alike 3.0 License](#). 

Portions not contributed by visitors are Copyright 2015 Tangient LLC