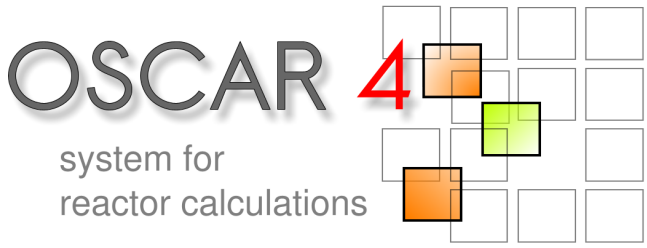
Doc. No. RRT-OSCAR-REP-09001
Revision No 1



OSCAR-4 Tutorial

Your First OSCAR-4 Run

RRT team: **Rian H. Prinsloo**

Last update: 6/09/2009

Radiation and Reactor Theory

Abstract

This tutorial acts as the first contact for new OSCAR users regarding hands-on use of the code, and aims to give an impression of the overall calculational path and the codes involved. A near complete reactor calculation is provided and the user is required to work through the basic steps, from generating cross-sections and parameterizing them, to finally applying the parameterized cross-sections to a simple reactor diffusion calculation. All input sets are provided, and the user simply has to execute the various codes during each step of the process. Some time is spent on viewing input and output, and on highlighting important file types and conventions.

Copyright

Legal Notice

Distribution Notice

Contact information

Postal address

Necsa

Building 1900, Radiation and Reactor Theory

P O Box 582

Pretoria

0001

South Africa

CompuSim AB

Strängnäsgatan 6

SE-723 34 Västerås

Sweden

Tel.:

+27 (0) 12 305 5042

+46 (0)21 413262

e-mail:

oscar4@necsa.co.za

info@compusim.se

Contents

1 Introduction **6**
- 1.1 Background . . . 6
- 1.2 The Scenario . . . 7
- 1.3 The OSCAR Approach . . . 7

2 Step-by-Step **10**
- 2.1 Getting Started: The OSCAR Setup for this Scenario . . . 10
- 2.2 Step 1: Generating Cross-Sections for an MTR Fuel Element . . . 11
- 2.3 Step 2: Parameterizing the Homogenized Cross-Sections . . . 12
- 2.4 Step 3: Creating a Linked Cross-Section Library . . . 13
- 2.5 Step 4: Performing the Core Diffusion Calculation . . . 13

3 Exercises **15**

4 Review of Activities **16**
- 4.1 Discussion . . . 16
- 4.2 Quiz . . . 17

5 Conclusion **18**

6 Solution to Quiz **19**

Symbols and Abbreviations

HEADE HEterogeneous Assembly DEpletion code

HEU High Enriched Uranium

LINX Cross-section library linking code

MANM Multigroup Analytic Nodal Method

MGRAC Multi-Group Reactor Analysis Code

MTR Material Testing Reactor

OSCAR-4 Overall System for the CAlculation of Reactors, generation 4

POLX Polynomial cross-section fitting code

1 Introduction

Contents

1.1 Background 6
1.2 The Scenario 7
1.3 The OSCAR Approach 7

Welcome to your first hands-on encounter with the OSCAR-4 code system. OSCAR-4 is a reactor calculational system, with specific focus on (in this generation of the system) research reactor calculational support. With this in mind, this tutorial series will aim to:

- Prepare you for using the code system for typical research reactor calculational applications.
- Provide you with sufficient background and examples to discover the rest for yourself.

In this specific tutorial, we will walk through the major calculational steps involved in preparing and executing research reactor calculations. The steps and codes involved in this process may seem numerous, but they follow the standard approach of utilizing deterministic transport solvers to homogenize cross-sections, and diffusion methods for full core modeling with depletion.

Although this approach represents standard practice for light-water moderated power reactors, some adaptations are required for the small heterogeneous reactor cores often found in research reactor designs. We will learn more about these techniques in future tutorials.

Tip: The typical improvement required is the use of multi-group diffusion theory, often 7 or more energy groups.

For the moment, let us focus on the joys of instant gratification, and perform our first reactor simulation via a few simple steps...

1.1 Background

Generally, three steps are involved in reactor simulation:

- The generation of homogenized few-group cross-sections for your reactor components, for the appropriate range of conditions (states) needed during actual full core simulation. This process implies the definition of the set of state parameters which will be varied to produce a set of homogenized few-group cross-sections for all possible future states of the fuel assembly.
- The set of homogenized cross-sections are fitted in order to produce a continuous representation of the few-group data.

- The reactor core is assembled from these homogenized materials, and a reactor operational cycle is defined and simulated, allowing the prediction of important reactor parameters like k_{eff}, power distribution and material number densities. At this point some terminology should be clarified. The module/code used to calculate diffusion solution on the scale of the full reactor problem is often refered to as the *global solver*, or alternatively the *core simulator*. These terms are used interchangeable throughout the tutorial series.

After this tutorial, you will be able to...

- ☐ **Identify which codes exist in the OSCAR-4 suite.**
- ☐ **Run the basic codes and understand their basic function.**
- ☐ **Identify and interpret the basic input and output files in OSCAR-4.**

1.2 The Scenario

The nuclear engineer in charge of performing reload calculations at your institution complains that his OSCAR installation is faulty. You need to perform a quick check of all the codes to ensure that they are functioning, consistent with one-another, and that no general problems exist regarding errors or warnings which the engineer should have been aware of. For this purpose, you will design and test the codes on a special "One Node Reactor" - a small reactor containing a single fuel element. You are required to re-generate and parameterize the fuel cross-sections for this simple reactor design, and finally to perform a core calculation, with depletion, using the global solver. It is convenient to know that for your one-node reactor, the result of the detailed transport calculation at the start of the process, and the homogenized diffusion calculation at the end of the process, should be the same so far as reactivity and system average behaviour is concerned. This is only true if equivalence parameters are generated and passed from the hetergeneous transport to the homogeneous diffusion calculation, as is the case for this calculation. This will be your criteria for success in testing the installation and consistency of the OSCAR-4 system.

1.3 The OSCAR Approach

In the OSCAR-4 code system, a number of different codes are used to address the problem of reactor calculations. The first step is the generation of few-group homogenized cross-sections. This step is typically performed via the HEADE code. HEADE is a collision probability transport solver, utilizing a response matrix formalism to solve a 2D fine-group transport problem for a given assembly type. More may be read about the HEADE code in some publications and user manuals, which may be found via the *OSCAR Document Navigation System*. The transport calculations are performed for a wide range of state conditions of the assembly, with typical

parameters being fuel temperature, moderator temperature, moderator density and burnup. Other parameters like the boron concentration, the Xenon-level and the burnable poison fraction are available in principle.

The HEADE code allows you to vary all of these within a single run, and dumps the discreet set of homogenized cross-section data, generated by utilizing the transport solution as a weighting function for both spatial homogenization and energy group condensation. The result of using this code, is the creation of a HED file.

Tip: Other few-group cross-section generation codes in OSCAR-4 are EQUIVA and STYX, but more about them later.

At this point the POLX code is used. POLX fits multiple quadratic polynomials through the few-group homogenized cross-sections in order to produce a continuous representation of the data. Read more about POLX and its methods in the User Guides, Theory Manuals and Related Papers sections of the *OSCAR Document Navigation System*. The result of this step is to produce a PLX file for every material mixture which you will later utilize in the global solver.

The final step in cross-section preparation requires the use of the LINX tool. LINX is a multipurpose cross-section library creation tool, with the specific function to merge PLX files for each assembly type in the reactor together into a single run-time cross-section library or LNX file. This file is later read by the global diffusion solver MGRAC. MGRAC is the primary workhorse of the OSCAR-4 system, since it is used to calculate all reactor parameters for every cycle simulation. MGRAC performs global multi-group nodal diffusion and depletion calculations, using the Multigroup Analytic Nodal Method (MANM) as the primary solution method. The code requires a number of input files, all of which are discussed in detail in future tutorials. The structure of the code system is illustrated in Figure 1.1.

Tip: Typically the library (LNX file) is created once, and used for many cycle simulations, until a new assembly type is introduced in the reactor.

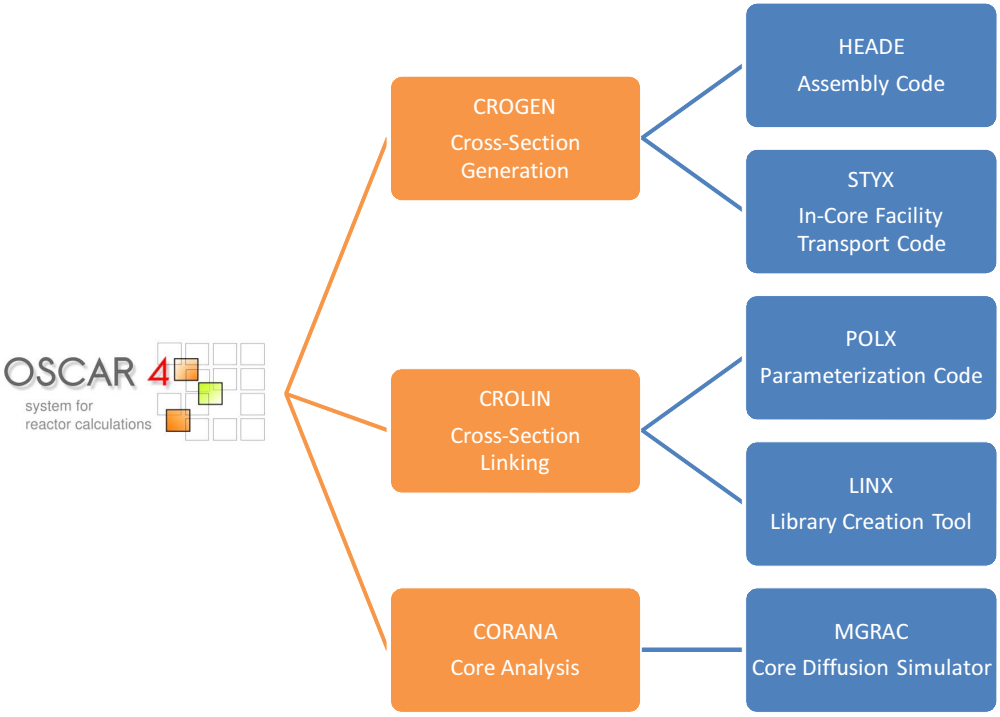


Figure 1.1: Structure of the OSCAR-4 Code System

2 Step-by-Step

Contents

2.1 Getting Started: The OSCAR Setup for this Scenario 10
2.2 Step 1: Generating Cross-Sections for an MTR Fuel Element 11
2.3 Step 2: Parameterizing the Homogenized Cross-Sections 12
2.4 Step 3: Creating a Linked Cross-Section Library 13
2.5 Step 4: Performing the Core Diffusion Calculation 13

We now proceed in a step wise manner and execute HEADE, POLX, LINX and MGRAC. This process will ensure that your installation is correct, and that you are capable of executing these codes with the input examples provided.

2.1 Getting Started: The OSCAR Setup for this Scenario

In order to proceed with this tutorial, OSCAR-4 should be installed on your system (see *Installation Tutorial*). Furthermore, the tutorial documents we will work through in this OSCAR-4 tutorial series, use examples and semi-completed inputs from a number of different reactors to illustrate various aspects of the code system. These reactor specific input and output files may be found at (in this case for the *One Node Reactor*):

```
<path>/files/One_Node_Reactor/crogen/
<path>/files/One_Node_Reactor/crolin/
<path>/files/One_Node_Reactor/corana/
```

where the three different paths refer to cross-section generation, cross-section linking and core analysis, respectively.

`<path>` is a `path` to your OSCAR-4 Tutorial Series - and is typically set to:

```
/home/oscar4/documents/Navigate/Tutorial/Tutorial_System
```

The codes within the OSCAR-4 suite should also be available in your system `PATH` environment variable (set by the installation procedure). We will use most of them in the following step-wise approach to your first OSCAR-4 run.

Please note that `xemacs` is used within this tutorial system as the default text editor. Please feel free to use any editor with which you are comfortable, or click to the link provided to open the document in the default associated editor in your system.

2.2 Step 1: Generating Cross-Sections for an MTR Fuel Element

Discreet sets of few-group cross-section data, dependent on various values of the state parameters, are generated by the HEADE code. Now you will execute HEADE for the first time. Before you do, quickly browse the HEADE input file for your fuel element. The element is a Material Testing Reactor (MTR) plate type (19 plates), 300 g High Enriched Uranium (HEU) fuel element.

To view the HEADE input file of the fuel element which we will place in our *One Node Reactor* go to:

```
>cd <path>/files/One_Node_Reactor/crogen/
>xemacs Tut_FirstRun_heade_start.INP
```

Click to Open

Inspect this file, and browse through the HEADE user manual. A full tutorial on the use of HEADE will follow later, but for now note the basic structure of a HEADE input file, with data describing, amongst others:

Tip: Remember that HEADE requires a complete heterogeneous description of the target assembly, in order to appropriately homogenize it with an energy and spatially detailed flux profile.

- Generic calculational data
- Heterogeneous fuel element description
- Burnup-up data
- Offbase ranges and state space definition

Now you are ready to perform your first HEADE run. This process will take a few minutes since the code performs multiple transport calculations on the given geometry for different values of burnup, fuel temperature, moderator temperature and moderator density. It is also important to note that HEADE will not run if the `HED` file is present in the directory. Make sure the `Tut_FirstRun_heade.HED` file is not in the directory, and then type:

```
>heade Tut_FirstRun_heade_start.INP
```

and go and find a cup of coffee...

The result of the HEADE run is the production of two files: `Tut_FirstRun_heade.PRT` and `Tut_FirstRun_heade.HED`. The latter is the specific format in which the homogenized cross-sections is passed to the next step in the calculational path, namely parameterization and the POLX code. The `PRT` file is a printout file for the run, in which all kinds of diagnostic information, warnings and errors may be found. If you wish to rerun the HEADE calculation at any time, always remove the `HED` output file first. Now you will be assigned your first task:

Task: Open the PRT file and check for errors and warnings in your calculation. Determine the initial infinite multiplication factor k_∞ of the fuel element and keep the value for future comparison to the *One Node Reactor* diffusion calculation. (Hint: Search the printout file for the "K(inf)")

2.3 Step 2: Parameterizing the Homogenized Cross-Sections

In the previous step, the HEADE code was used to calculate homogenized diffusion parameters for a fuel element. These homogenized cross-sections were calculated for various potential reactor conditions (such as burnup, fuel temperature, moderator temperature and moderator density) and stored as discreet cross-section sets in the HED file. You are now required to present this data in a continuous form for later use in the global diffusion calculation. For this purpose, we will apply the POLX code, which fits the cross-section data as a function of the four state parameters, within a given tolerance, utilizing quadratic polynomials. Now we will execute the POLX code together, and fit the cross-sections which you produced in the previous step. Before we execute the POLX code, open the input file and acquaint yourself with the basic input data by referring to the POLX user guide. Open the input file:

```
>cd <path>/files/One_Node_Reactor/crolin/
>xemacs Tut_FirstRun_polx_start.INP
```

Click to Open

Notice that the POLX input is relatively simple, and that most of the data needed for the run is passed to the code via the HED file. Now perform the POLX run:

```
>polx Tut_FirstRun_polx_start.INP
```

The calculation may typically take a few minutes depending on the convergence criteria and the number of points to fit, but for this simple example with no-offbases (depletion performed at base conditions for values of all state parameters) and a handful of burnup steps, it should finish in a couple of seconds. The run will produce Tut_FirstRun_polx.PLX which contains the polynomial coefficients for the fitted cross-sections of the fuel element.

Task: The default tutorial calculation ran with a fitting tolerance of 0.1%. Repeat the POLX run, but this time with the convergence criteria set to fit the cross-section to within 0.01% accuracy. Keep in mind that this criteria directly impacts upon the accuracy of the core calculation later on. Continue with the rest of the tutorial using the refined POLX run.

2.4 Step 3: Creating a Linked Cross-Section Library

Eventhough our reactor contains only a single material (homogenized fuel), a typical cross-section library for a core calculation could contain hundreds of mixtures, each represented by polynomial coefficients stored in a separate PLX file. It is customary to combine the files into a single cross-section library, containing all cross-sections needed for global reactor calculations. This is done via the LINX tool, which reads a number of PLX files and combines them into a single LNX file. Open the LINX input file now and have a look at the basic input data provided:

```
>xemacs Tut_FirstRun_linx_start.INP
```

Click to Open

Note that the list of PLX files are not explicitly given in the LINX input file, but rather only a cryptic reference to a file POLX.DIR. You will learn more about this file later on, but suffice to say that POLX generates and maintains this file to contain all mixtures which have passed through the POLX code. This eases your input preparation for LINX substantially. Now it is time to prepare your LNX library, by executing the following command:

```
>linx Tut_FirstRun_linx_start.INP
```

2.5 Step 4: Performing the Core Diffusion Calculation

The final step in your first complete OSCAR-4 run requires the use of the MGRAC code, and the execution of a full-core global diffusion calculation. Typically this calculation would require a full core assembly layout to be defined, constructing of various assembly types based on the cross-section mixtures available in the LINX library. For our *One Node Reactor*, this process will be substantially simplified, with a simple one node core defined, containing only one assembly type, and of course only one element of this type. To see how this scenario is communicated to the MGRAC code, browse through the following files prior to running the code. Five different input files are involved:

- ▷ CONFIG file - describes the core geometric layout
- ▷ LOAD file - describes the placement of the elements prior to the start of the cycle
- ▷ BASE file - describes the axial material structure of a given base assembly type
- ▷ HIST file - describes the isotopic data of an actual, named assembly
- ▷ INPUT file - describes the cycle/scenario to be simulated - hence power levels, rod positions, depletion steps, etc.

Browse through (or click the links above to see) the user guides for each of the input file types. To view the actual files for your *One Node Reactor*, enter:

```
>cd <path>/files/One_Node_Reactor/corana/
>xemacs Tut_FirstRun_mgrac_start.CONFIG  -- core layout file
```

```
>xemacs Tut_FirstRun_mgrac_start.LOAD     -- core load file
>xemacs Tut_FirstRun_mgrac_start.BASE     -- assembly base file
>xemacs Tut_FirstRun_mgrac_start.HIST     -- assembly history file
>xemacs Tut_FirstRun_mgrac_start.INP      -- mgrac input file
```

Open CONFIG Open LOAD Open BASE Open HIST Open INPUT

Now continue to run the MGRAC calculation, by entering:

```
>mgrac Tut_FirstRun_mgrac_start.INP
```

For a full cycle calculation, with a typical research reactor core layout, this calculation should take a few minutes, but for our very simple reactor, the calculation should only take seconds. The calculation should produce a printout file with all the result data. View this file by entering:

```
>xemacs Tut_FirstRun_mgrac.PRT
```

Near the end of the file you will find a simulation summary, where you can see the major results for each calculational case.

Task: Find within the “Simulation Summary” section of the printout file the calculated k_{eff} values for your core (one for each depletion step), and compare the initial value to the initial k_∞ you calculated during the HEADE calculation. These values should agree - can you explain why?

3 Exercises

The exercises you have performed in this tutorial required minor interpretation of the input and output files associated with the most important codes of the OSCAR-4 system. Hopefully you would have noticed that the k_∞ you produced in HEADE and the k_{eff} you produced in MGRAC agree to within convergence criteria. This is of course due to the correct implementation of equivalence theory [1] in the OSCAR system, although this infinite medium single assembly scenario does not test the generated equivalence parameters fully. You can read more about the OSCAR-4 system's predecessor OSCAR-3 (and some its applications) in [2], or see a more extensive list of related OSCAR-4 publications accessible from the *OSCAR Document Navigation System*.

Exercise 1

As an additional exercise, compare the full range of burnup-dependent k_∞ values in HEADE to the burnup dependent k_{eff} values in MGRAC. Do you notice any discrepancies, and if so, can you explain them?

4 Review of Activities

Contents

4.1 Discussion 16
4.2 Quiz 17

4.1 Discussion

Firstly, it is important to note that this tutorial focused on showing the typical OSCAR-4 calculational path, and not specifically how to calculate a meaningful problem. Nevertheless, we firstly performed a transport solution for the heterogeneous fuel assembly problem, and used this solution to homogenize and condense the assembly into a single set of nodal diffusion equivalence parameters (node-averaged cross-sections and discontinuity factors). Thus, the concept of the equivalence of heterogeneous transport and homogeneous diffusion solutions has been illustrated in this tutorial, and provide a base for meaningful testing and understanding of the components of OSCAR-4. Of course, in a real reactor calculation, the global diffusion problem would contain a number of assemblies placed within configurations somewhat different from the idealized infinite medium lattice applied in the HEADE calculation, and hence, equivalence is partially lost. The basic assumption in this approach, however, is that the variation of the core environment from the lattice environment, and its associated impact on the node-averaged cross-sections, is small. Nevertheless, the methodology applied in representing the few-group cross-sections should capture many of these environmental, or spectrum, effects. You will learn more about this rather complex topic as the tutorial series progresses.

In the exercise, which required you to compare the burnup dependent reactivity predictions from HEADE and MGRAC, some differences would exist even for the idealized problem in this tutorial. These differences are mainly due to:

- Differences in burnup methods and isotopic chains applied in HEADE and MGRAC (mitigated by smaller burnup steps).
- Errors in the polynomial fitting of the node-averaged cross-sections (mitigated by tighter conversion criteria).

4.2 Quiz

The following set of questions will test some of the material covered in this tutorial. Please click on *Begin answering the quiz* to start the tutorial, select your answers, and then, when you are done, click *Check my answers* to see your score. After completing the quiz you may select the *Show correct answers* button to show the model answers, and then click on the correct answers to see the expanded solutions. Alternatively, scroll to the end of the document to see the expanded solutions, but do try the tutorial first...

begin answering the quiz

1. Which methods are applied in OSCAR-4 to solve reactor problems?

Transport | Diffusion | Both | Monte Carlo

2. Which code is used to fit the few-group homogenized cross-sections?

LINX | STYX | POLX | MGRAC

3. Which code in the OSCAR-4 suite uses the Multigroup Analytic Nodal Method (MANM) to perform the global diffusion solution?

CORANA | LINX | MGRAC | CROGEN

4. Which of the following input file types would describe axial material layout, in terms of homogenized cross-section mixtures, for a given reactor component?

`BASE` | `HIST` | `LOAD` | `CONFIG`

5. Which of the following input file types would describe the isotopic content of a specific burnable element at various stages during its lifetime?

`BASE` | `HIST` | `LOAD` | `CONFIG`

6. What is the suggested number of broad energy groups to be used in the modeling of small research reactor cores?

1 | 2 | 4 | 6 or more

check my answers

5 Conclusion

Congratulations on having completed your first OSCAR-4 tutorial! Eventhough the problem you solved seems far removed from a real practical reactor calculation, you have learned how OSCAR-4 operates and how to approach the problem. Furthermore, this tutorial provides a simple and easy way to check whether your installation is functioning and whether all the codes in your current release are compatible with one-another. You are now ready to move on to a more in-depth look at the various codes in the OSCAR-4 suite, and to slowly work your way through to performing full reactor safety, utilization and core-follow calculations. Good luck and enjoy the rest of the tutorials. You should now move onto the next tutorial in the series, *Adding a New LEU Fuel Type to the Core - Part 1*.

6 Solution to Quiz

Solution: Both. HEADE, amongst other codes, applies transport theory to produce homogenized and condensed cross-sections, which are passed to MGRAC in order to solve the global reactor diffusion problem. back to quiz ◄

Solution: The POLX code applies a second order least squares fit to the state parameter dependent set of few-group cross-sections, in order to produce a polynomial representation of the data. back to quiz ◄

Solution: MGRAC is an acronym for the Multi-Group Reactor Analysis Code and allows the calculation of full-core full cycle calculations in a couple of minutes. back to quiz ◄

Solution: The `BASE` file describes which mixtures, of those available in the cross-section library (`LNX` file), are used to axially construct a given reactor component type. back to quiz ◄

Solution: The `HIST` file containes a description of the time stamped isotopic content of a specific element for all material layers within the element. A `HIST` file exists for every element, while a `BASE` file exists for every element type. back to quiz ◄

Solution: Typically, 6 or more groups are suggested, due to the strong impact of the leakage spectrum upon the core behaviour back to quiz ◄

Bibliography

[1] Smith, K. S. “Assembly Homogenization Techniques for Light Water Reactor Analysis”, *Progress in Nuclear Energy*, 1986, *17*, pp. 303-335. 15

[2] Reitsma, F.; Joubert, W. R. “A Calculational System to Aid Economical Use of MTRs”, *International Conference on Research Reactor Fuel Management (RRFM’99)*, Brugges, Belgium, 29-31 March 1999.

15