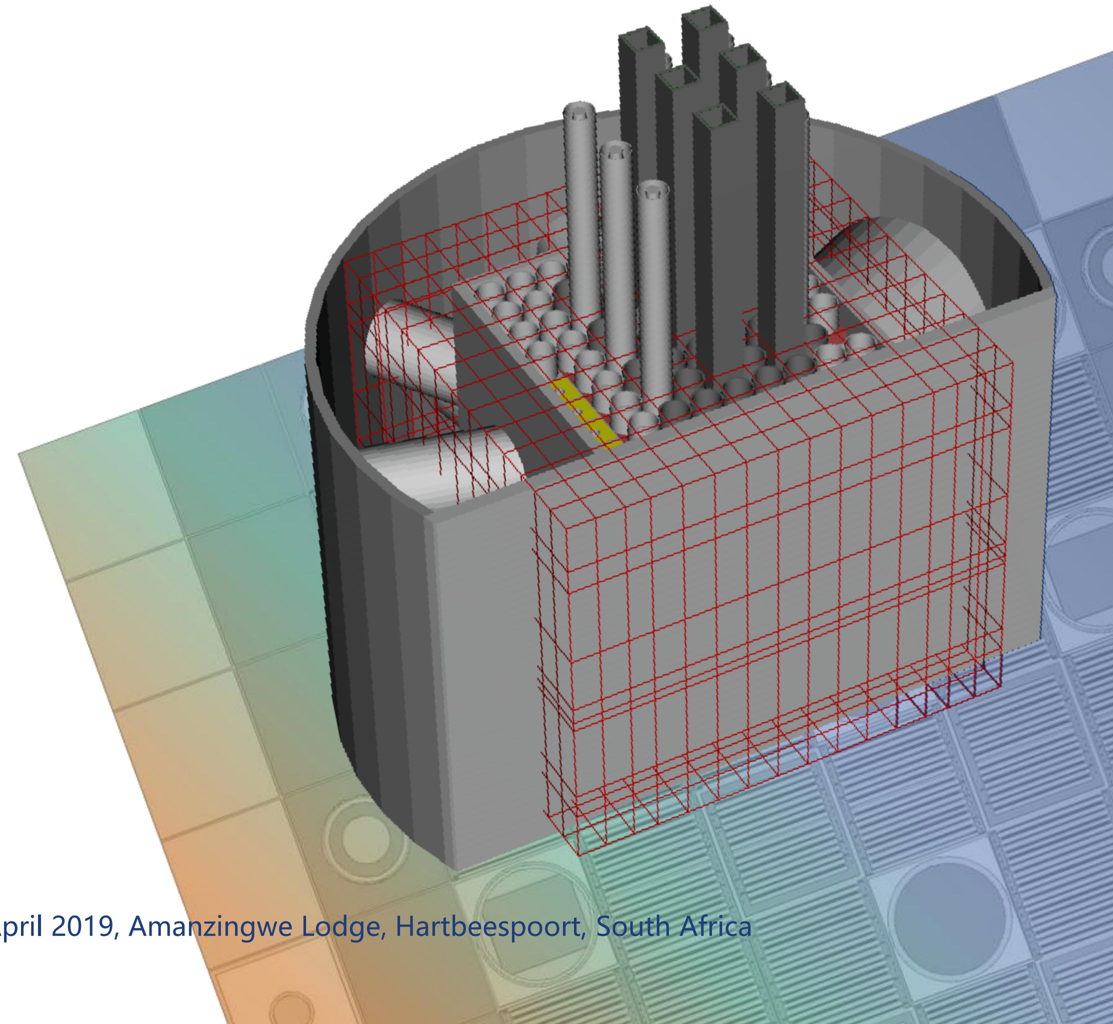
Core-follow and related applications
Full-core case

OSCAR-5 Tutorial

Presenter:
Rian Prinsloo



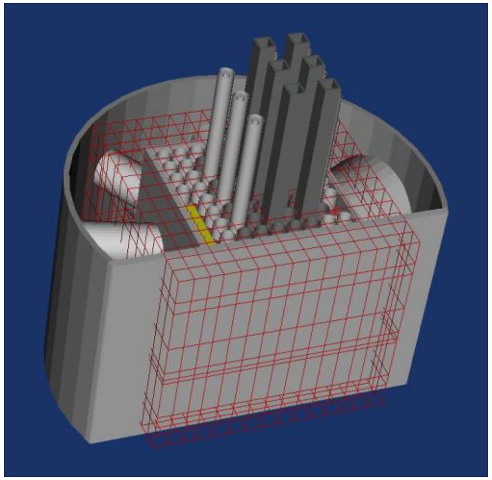
OSCAR-5 REACTOR ANALYSIS PLATFORM

necsa We're in your world

OSCAR-5 Release and User Group Workshop, 15 – 19 April 2019, Amanzingwe Lodge, Hartbeespoort, South Africa
Hosted by Radiation and Reactor Theory, Necsa

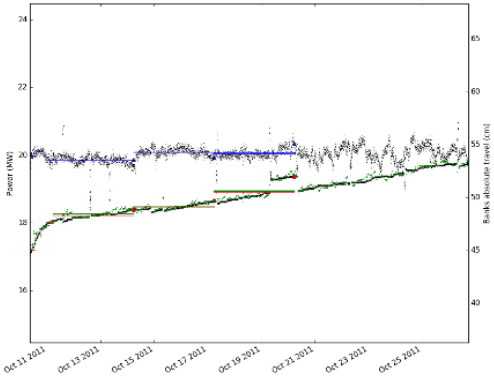
Outline

- Background
  - Introduction
  - What was done so far
  - What you will do now
  - The scenario



- The OSCAR Approach

- Ready, set, go!



- Conclusion

Background

- This tutorial is about typical calculational applications

- What you have done so far
  - Tutorial 1: See the overall workflow
  - Tutorial 2 part 1: Build a heterogeneous model
  - Tutorial 2 part 2: Build a homogeneous model

- Focus in Tutorial 3:
  - Introduce the OSCAR-5 benchmark: SAFARI-1 core model
  - Creating a starting condition at a given cycle
  - Perform reload analysis
  - Perform core-follow analysis

Background

What you will learn now

1
- Initiate the inventory, from an existing data source and load the facility

2
- Perform reload analysis for the coming core – safety and utilization parameters and generate reload report

3
- Process plant data and perform for core-follow analysis

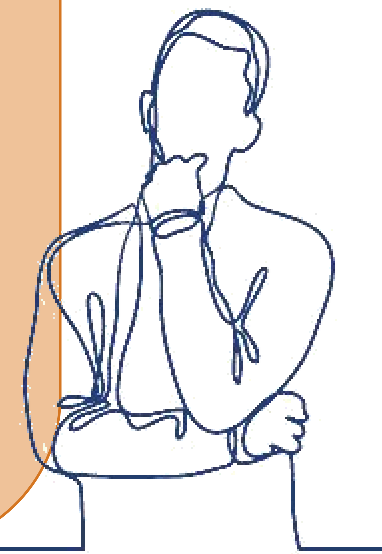
4
- Generate a high-fidelity core-follow line

Background

The scenario

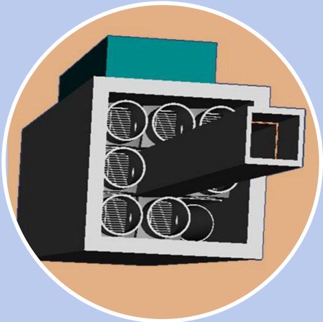
You are in the process of migrating your reactor simulation tools from OSCAR-4 to OSCAR-5. You have a functioning OSCAR-4 corefollow set and you have just created a full core nodal model with OSCAR-5. Now you want to create a starting condition for your new OSCAR-5 models, and model your first reactor cycle in OSCAR-5.

You will need a starting fuel inventory for your model (to be taken from your OSCAR-4 model as example in this case) as well as a planned core loading for the coming cycle. Finally, you would need plant data from the reactor once the new cycle has finished its operation to model to fuel depletion in the core. After performing the reload analysis and subsequent core-follow calculation for the cycle, confirm the accuracy of the nodal model by running an independent core-follow calculation in Serpent for the same cycle.

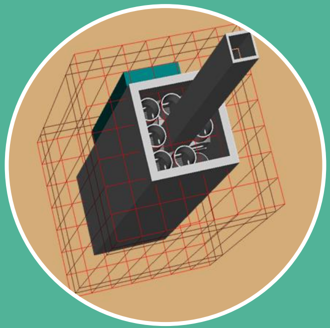


The OSCAR workflow

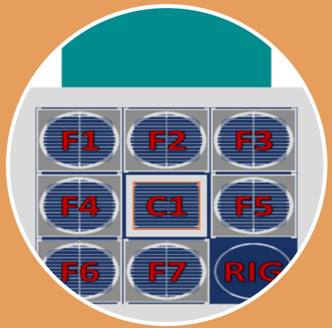
model



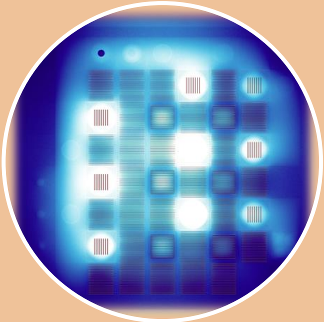
configuration



compose



facility loading

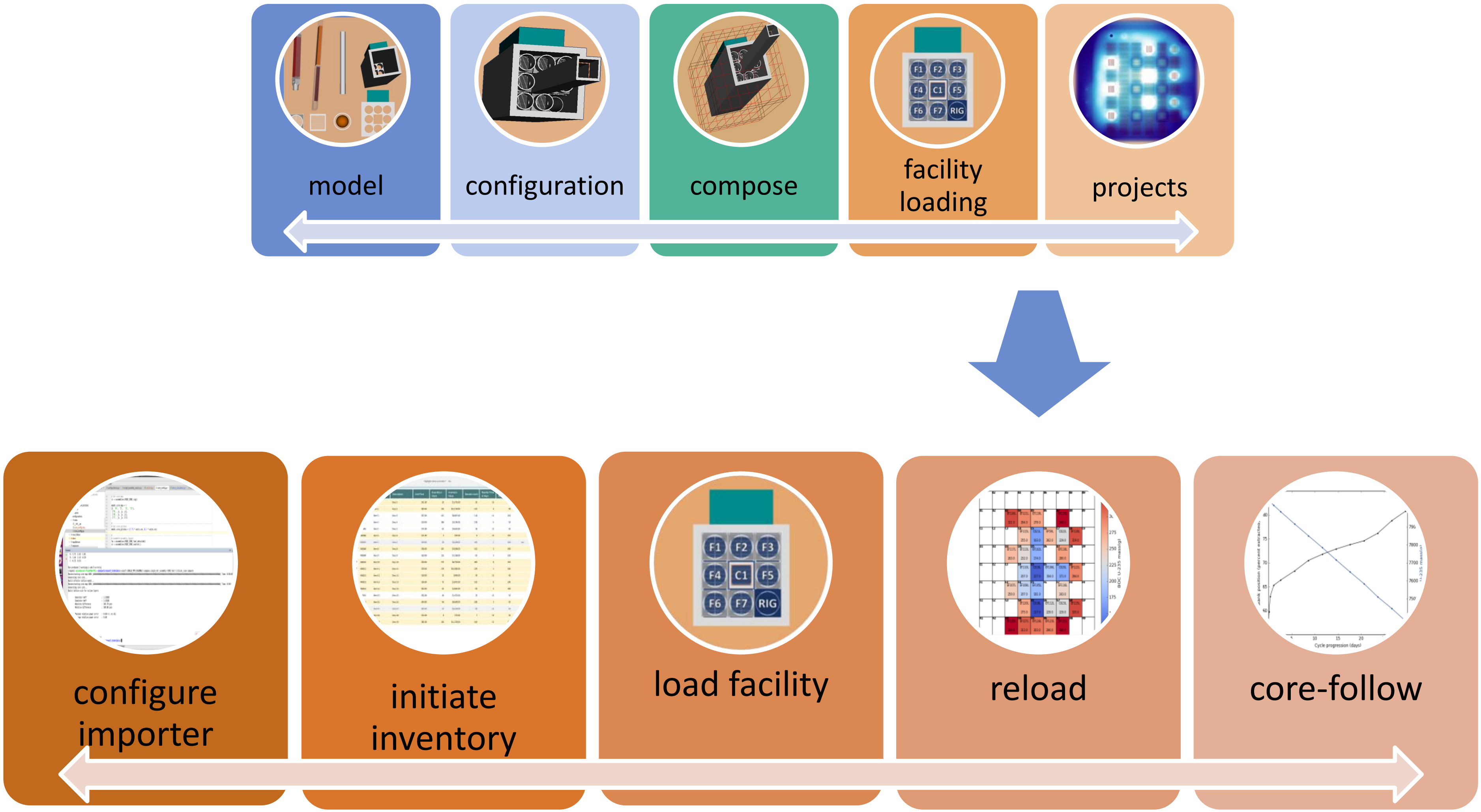


projects

The OSCAR Approach

- Initiate the burnable inventory with historic elements
  - Identify a data source: Example is historic OSCAR-4 set
  - Develop input file for inventory import
- Load the core facility for a given cycle
  - Input file for the facility manager
- Reload analysis
  - Generic input file for structuring reload analysis calculations
  - Cycle-specific input file for cycle specific reload analysis
  - Customized report template for generating reload report
- Core-follow analysis
  - Process plant data, and generate a step-wise structure (Graphical process)
  - Develop input file for core-follow calculation
- High fidelity core-follow line
  - Same input file for core-follow application

The OSCAR Approach: projects



Ready, set, go!

Getting Started: The OSCAR-5 Setup for this Scenario

Step 1: Study the existing model documentation

Step 2: Obtaining a starting inventory

Step 3: Facility loading

Step 4: Perform reload calculations

Step 5: Perform core-follow calculations

Step 6: Creating a second independent core-follow line

Ready, set, go!

Getting started: Orientation and setup

- Browse through section 1 (covered in presentation)
- The OSCAR-5 setup for this scenario
- This tutorial takes place in the validation space (not the oscar5_tutorial space)
  - "tutorial_" preface
- How long should be spent on this step?
  - Spend ~10 min on this step

Ready, set, go!

Step 1: Studying the existing reactor model

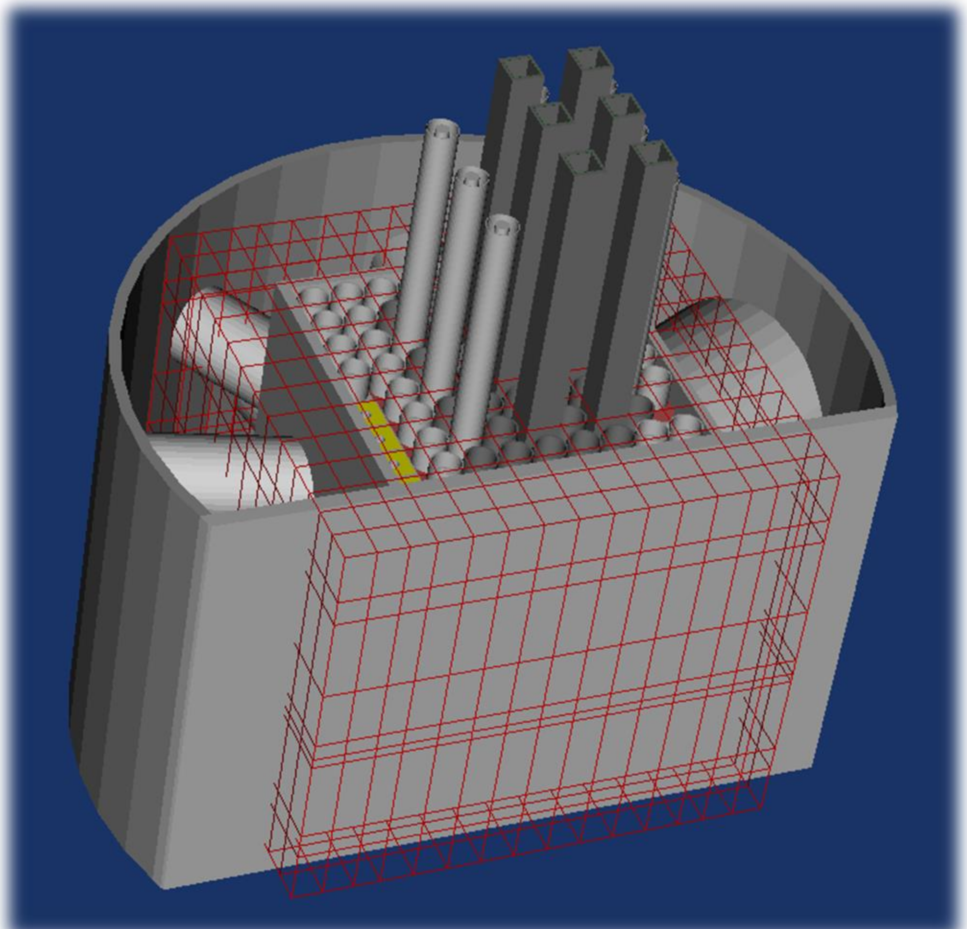
- What will you do?
  - Browse the auto-generated documentation of the SAFARI-1 model to familiarise yourself with the model
  - Look at OSCAR-5 input
    - Component models are built via MCNP input import
    - Look at the heterogeneous component and core designs
    - Look at the homogeneous model and strategy
  - Do not get lost in the details…

- What will you learn in this step?
  - What the SAFARI-1 core layout consists of

- How long should be spent on this step?
  - Spend about 15 minutes on this step

The result: Step 1

Getting started: The SAFARI-1 benchmark



On we go!

Step 2: Obtaining a starting inventory

- What will you do?
  - Obtain a starting inventory
  - Import the historic set of fuel elements with a burn-up history
- What will you learn in this step?
  - How is burn-up data imported and stored
  - How are residual isotopes treated in OSCAR-5
- How long should be spent on this step?
  - You should spend about 15 minutes on this step

Tasks for this step

Step 2: Obtaining a starting inventory

Initiate the inventory for the nodal code

oscar5 SAFARI_1.facility_loading.tutorial_initiate_inventory execute --force

Creates SAFARI_1/SAFARI_1_tutorial_inventory database

Initiate the inventory for the high-fidelity line

oscar5 SAFARI_1.facility_loading.tutorial_initiate_inventory execute --force

Creates SAFARI_1/SAFARI_1_tutorial_inventory_serpent database

The result: Step 2

Step 2: Obtaining a starting inventory

Databases updated

internal binary files: browse-able via python interface

On we go!

Step 3: Facility loading

- What will you do?
  - Browse the facility loading file and understand its contents
  - Execute the facility loading
- What will you learn in this step?
  - Load a facility at a given time with a load map
- How long should be spent on this step?
  - You should spend about 10 minutes on this step

Tasks for this step

Step 3: Facility loading

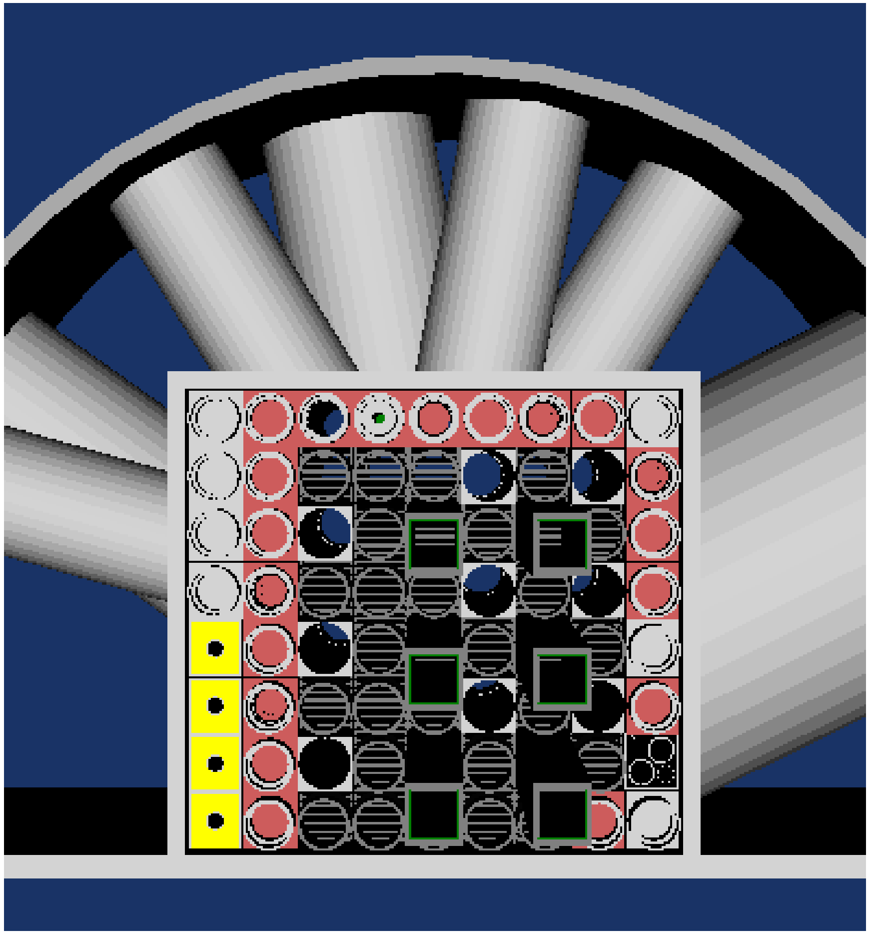
Load the core for C1109-1

```
oscar5 SAFARI_1.facility_loading.tutorial_C1109_1 execute --force
```

Updates SAFARI_1/SAFARI_1_facility_tutorial database

The result: Step 3

Step 3: Facility loading



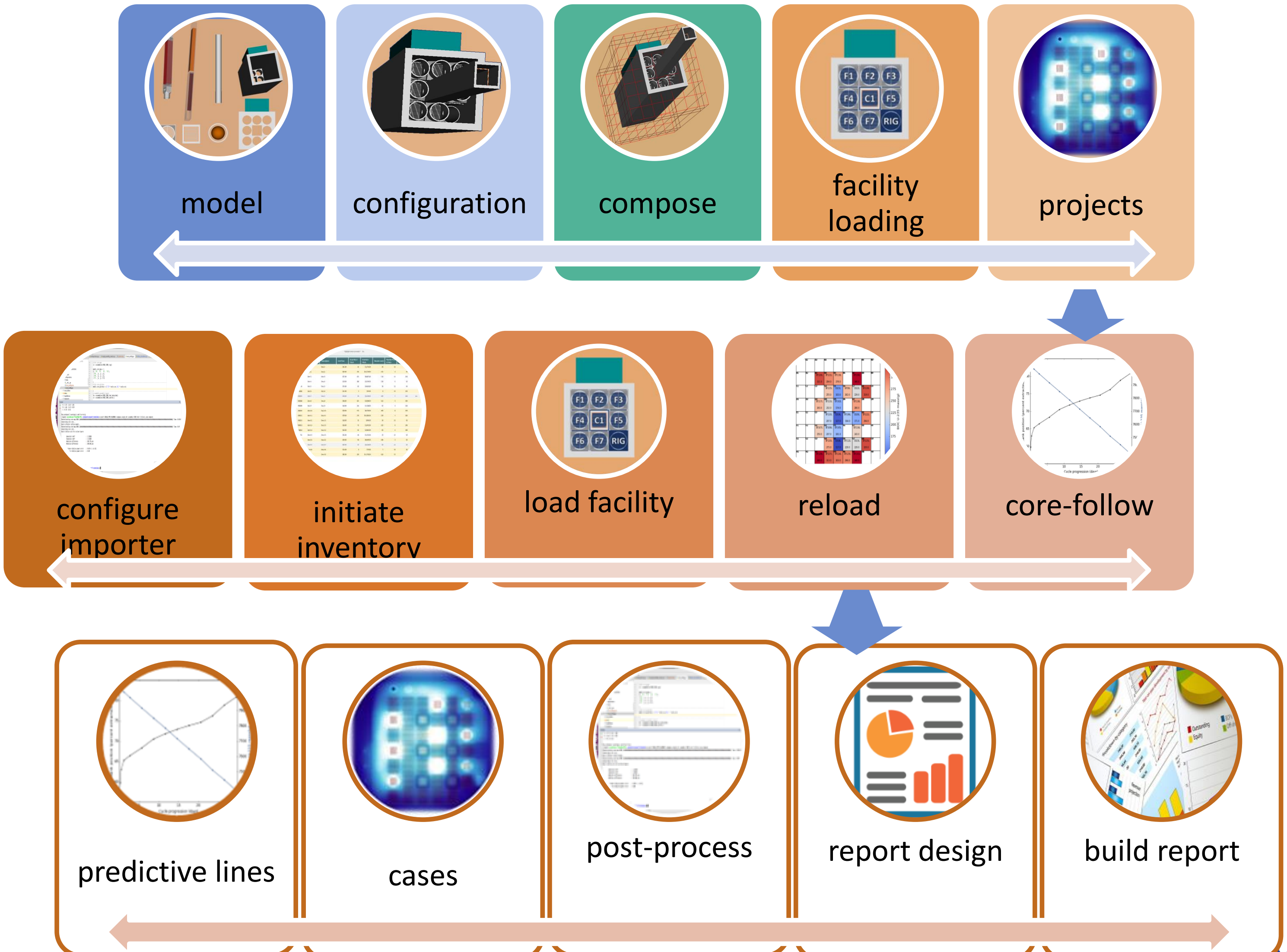
Databases updated

On we go!

Step 4: Perform a reload calculation

- What will you do?
  - Set up, perform and report on a reload calculation
- What will you learn in this step?
  - What is a reload calculation
  - What is the workflow of reload calculations
  - How to export the OSCAR-5 model to OASYS as alternative option
- How long should be spent on this step?
  - You should spend about 25 minutes on this step

The OSCAR Approach: reload report



Tasks for this step

Step 4a: Perform the predictive lines (all at once...)

Perform the predictive depletion calculations

oscar5 SAFARI_1.reload.tutorial_C1109_1 prediction --target-mode MGRAC execute --force

Generate a run series of MGRAC calculations in reload/C1109_1/PEAKING and reload/C1109_1/REACTIVITY

The result: Step 4a

Step 4a: Perform prediction line calculations

Input created and calculations run

Tasks for this step

Step 4b: Define specific additional results, or cases

Perform the predictive depletion calculations

```
oscar5 SAFARI_1.reload.tutorial_C1109_1 cases --target-mode MGRAC execute --force
```

Generate a run series of MGRAC calculations in reload/C1109_1/PEAKING_CRITICAL_CASE_0_0 and reload/C1109_1/REACTIVITY_BANK_WORTHS

The result: Step 4b

Step 4b: Perform additional result cases

Input created and calculations run

Tasks for this step

Step 4c: Produce reload report

Post process results reportable quantities

```
oscar5 SAFARI_1.reload.tutorial_C1109_1 --target-mode MGRAC post
```

Update tutorial_C1109_1/C1109-1_MGRAC.res files

Prepare the report (populate template with results)

```
oscar5 SAFARI_1.reload.tutorial_C1109_1 --target-mode MGRAC prepare-report
```

Populate reload/C1109_1/tutorial_C1109_1/report/source with html/latex files

Build the report

```
oscar5 SAFARI_1.reload.tutorial_C1109_1 --target-mode MGRAC  build-report
```

Build html and pdf report under reload/report/tutorial_C1109_1/report/build with source latex files

The result: Step 4b

Step 4c: Produce the reload report

reload/tutorial_C1109_1/report/build/html/index.html

On we go!

Step 5: Perform a core-follow calculation

- What will you do?
  - Process plant data for the past cycle
  - Generate a template core-follow input file
  - Perform the core-follow calculation
  - Browse the core-follow results
- What will you learn in this step?
  - How to set up and perform on a core-follow calculation
- How long should be spent on this step?
  - You should spend about 25 minutes on this step

Tasks for this step

Step 5a: Process the plant data

Perform the predictive depletion calculations

oscar5 SAFARI_1.core_follow.process_plant_data

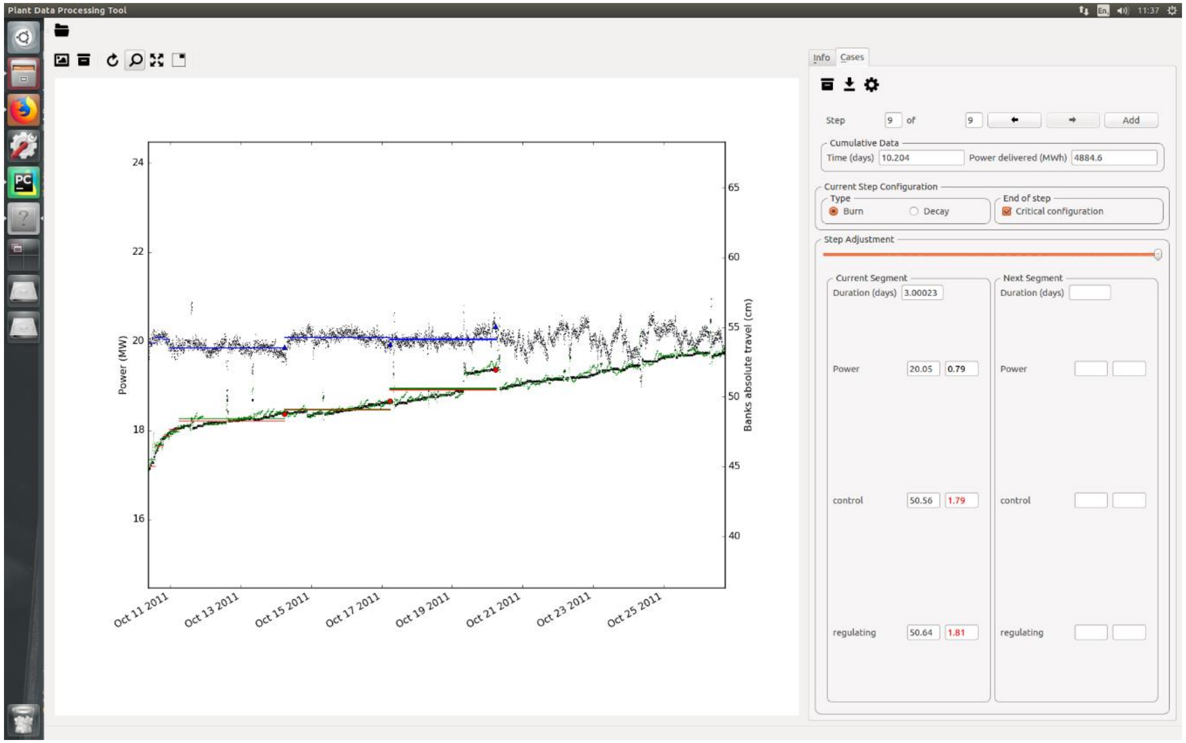
Open a GUI for introducing step structure

Save the step structure from the GUI

Save the C1109_1_plat_data.pkl file

The result: Step 5a

Step 5a: Process the plant data



Tasks for this step

Step 5b: Perform the core-follow calculation

Prepare the template input file

oscar5 SAFARI_1.manager cycle_corefollow TUT_C1109_1 --start-time ’ 10/10/2011 00:37:20’ --configuration base_hot_core

Create the corefollow/tutorial_C1109_1.py file

Execute the core-follow calculation

oscar5 SAFARI_1.core_follow.tutorial_C1109_1 --target-mode MGRAC execute --force

Generate MGRAC input and output under corefollow/tutuorial_C1109_1

Post process the results

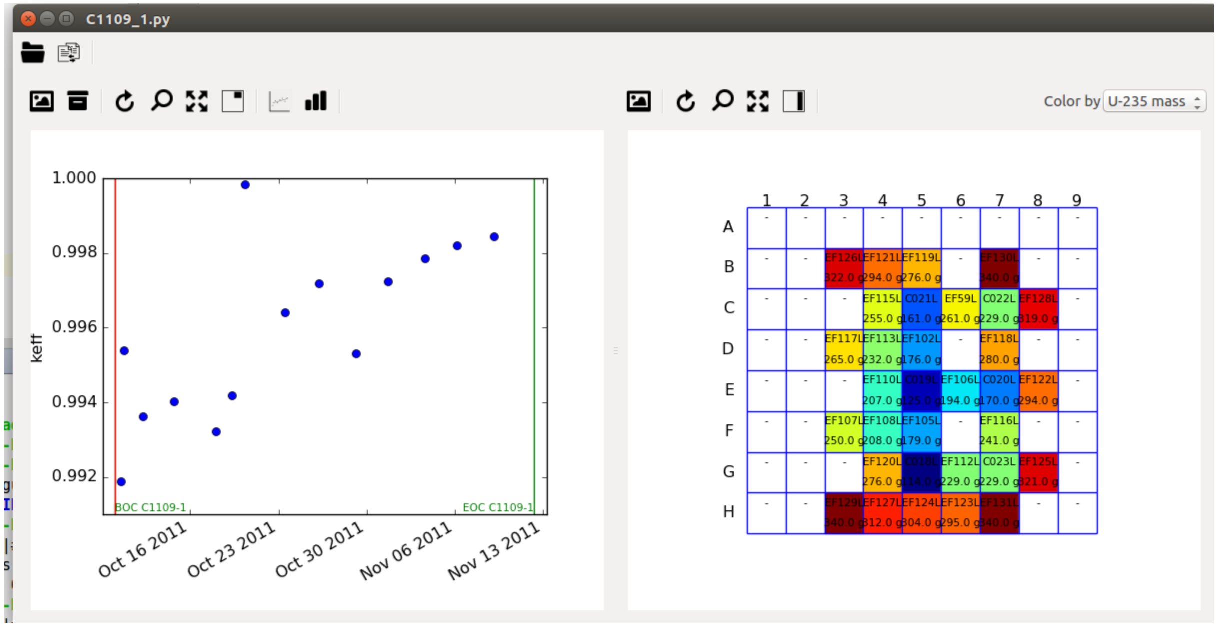
oscar5 SAFARI_1.core_follow.tutorial_C1109_1 --target-mode MGRAC post --show

Update the res file

The result: Step 5b

OSCAR-5
REACTOR ANALYSIS PLATFORM

Step 5b: Execute the core-follow calculation



necsa
We're in your world

On we go!

Step 6: Create an independent core-follow line

- What will you do?
  - Migrating your inventory
  - Repeat the core-follow calculation with Serpent
- What will you learn in this step?
  - Managing multiple inventories
  - Deploy a calculation to a high-fidelity independent line
- How long should be spent on this step?
  - You should spend about 25 minutes on this step

Tasks for this step

Step 6: Perform the core-follow calculation

Execute the core-follow calculation

```
oscar5 SAFARI_1.core_follow.C1109_1 --target-mode SERPENT execute --force
```

Generate SERPENT input and output under corefollow/tutorial_C1109_1

Post process the results

```
oscar5 SAFARI_1.core_follow.C1109_1 --target-mode SERPENT post --show
```

Update the res file

Conclusion

Congratulations!

- Congratulations on completing your last OSCAR-5 tutorial!

- What was done
  - Full-core application workflow – from initiate and inventory to reload and core-follow analysis with multiple codes

- What is next?
  - That's up to you….

Thank you