Development of an Advanced Nuclear Fuel Cycle Simulator (FANCSEE) with Graphical User Interface

B. Chmielarz (KTH, USNC Europe), W. Gudowski (KTH), Y. Hrabar (KTH), C. Ding (KTH, Tsinghua University),

J. Zou (KTH, Tsinghua University) and A. Bidakowski (Uppsala University)

9.07.2018 Paris, France

Contents

Introduction

- FANCSEE is a **standalone** advanced fuel cycle simulation code under development at KTH, Sweden

- Written in C and C++ for Linux

Idea behind FANCSEE

The software is meant to be:

- Graphically controlled
- User-friendly
- Suitable for implementation of complex scenarios
- Computationally efficient
- For researchers, policymakers and students

Technical aspects

- Compiled with gcc compiler
- Uses Cairo, GTK+ 2.4, Boost and standard C/C++ libraries
- Open-source software *Grace* is used for plotting
- Results can be exported to MATLAB files
- MATLAB is used for massive data processing to create compatible libraries from Serpent burnup

Technical aspects

- Uses the state-of-the-art Chebyshev Rational Approximation Method (CRAM) to solve transmutation
  - Method used in Serpent or ORIGEN 2.2+

$$\boldsymbol{n}(\Delta t) = a_0 \boldsymbol{n}(0) + 2Re\left[\sum_{i=1}^{k/2} (A\Delta t + I\theta_i)^{-1} a_i \boldsymbol{n}(0)\right]$$

  - The equation is solved for every $a_i \boldsymbol{n}(0)$ with Gaussian elimination, matrix $(A\Delta t + I\theta_i)$ does not need to be inverted

Main features

Reactor libraries:

- PWR (VVER440-213, reference NEA PWR) and BWR (ABB-III)
- Calculated with Serpent 2

Output:

- Tracking of up to 1346 isotopes over time
- Plotting of results for each facility and nuclide
  - Nuclide mass
  - Inhaled or ingested toxicity
  - Radioactivity

Scenarios

Scenarios are simulated through setting up **facilities** with discrete **functions** and **parameters**. There are general and facility-specific parameters.

Adjustable parameters of a scenario:

- Burnup timestep
- Report timestep
- Beginning and end of a scenario

Facility parameters

The possible facilities are:

- Uranium Mines
- Enrichment Plants
- Reprocessing Plants
- Fuel Factories
- Reactors
- Waste Repositories

Start and stop dates are adjustable for each facility

Facility parameters

- Mine, Enrichment Plant, Fuel Factory and Reprocessing Plant can have a processing capacity limit (in kg/day)
- Reprocessing Plant parameters
  - Reprocessing order
    - First In First Out (FIFO) or Last In First Out (LIFO)
  - Reprocessing limit – number of times a fuel batch can be reprocessed
  - Minimum cooling time before reprocessing [years]
  - Maximum viable age for reprocessing [years]

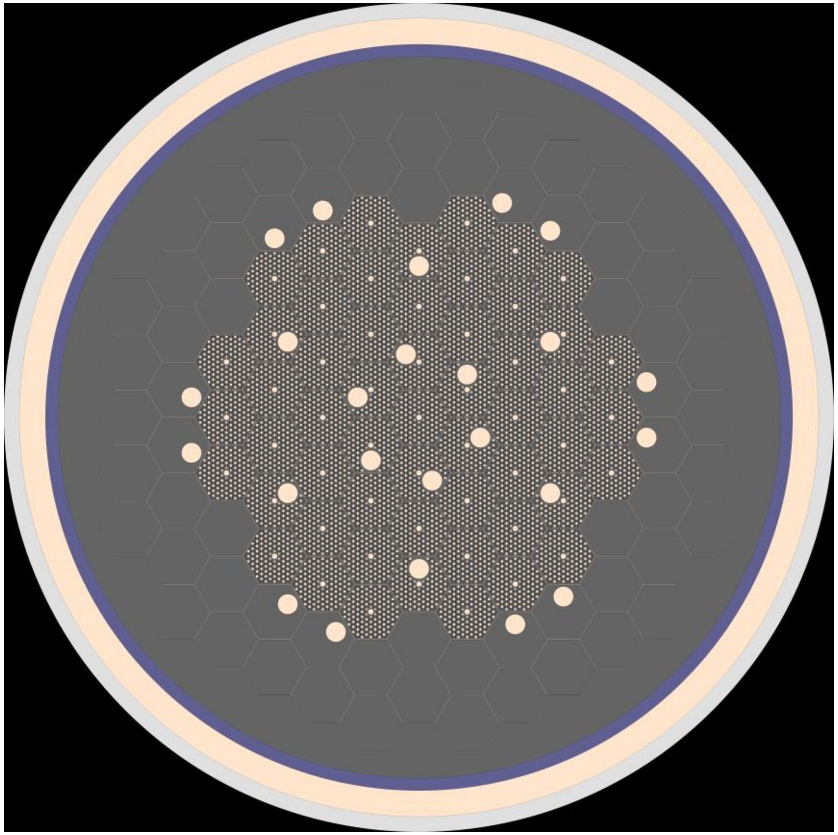
Software presentation

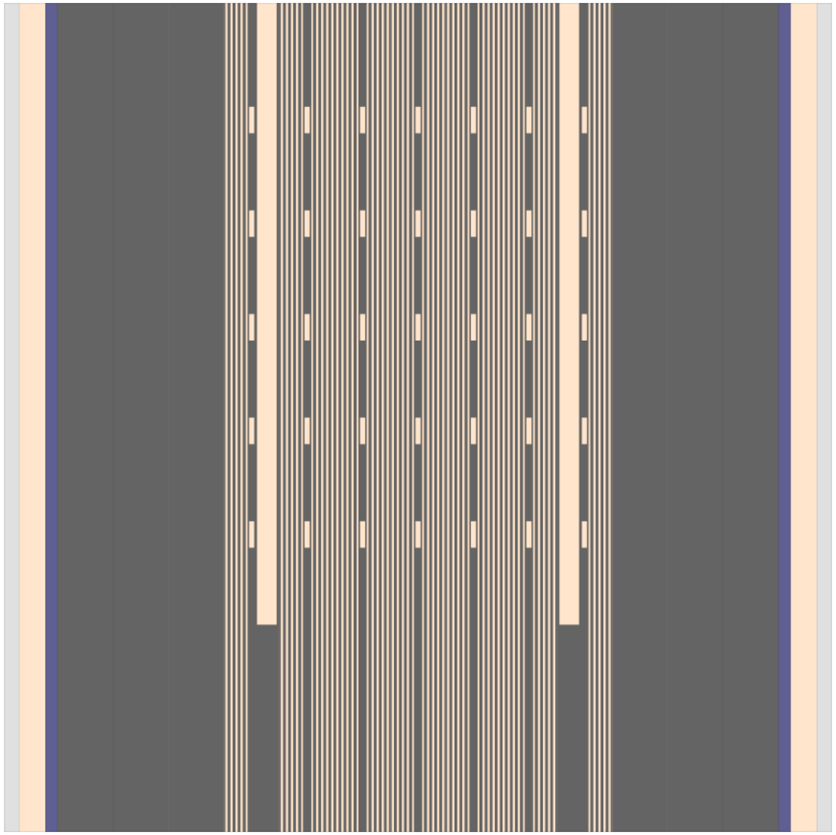
Current work

New reactor types:

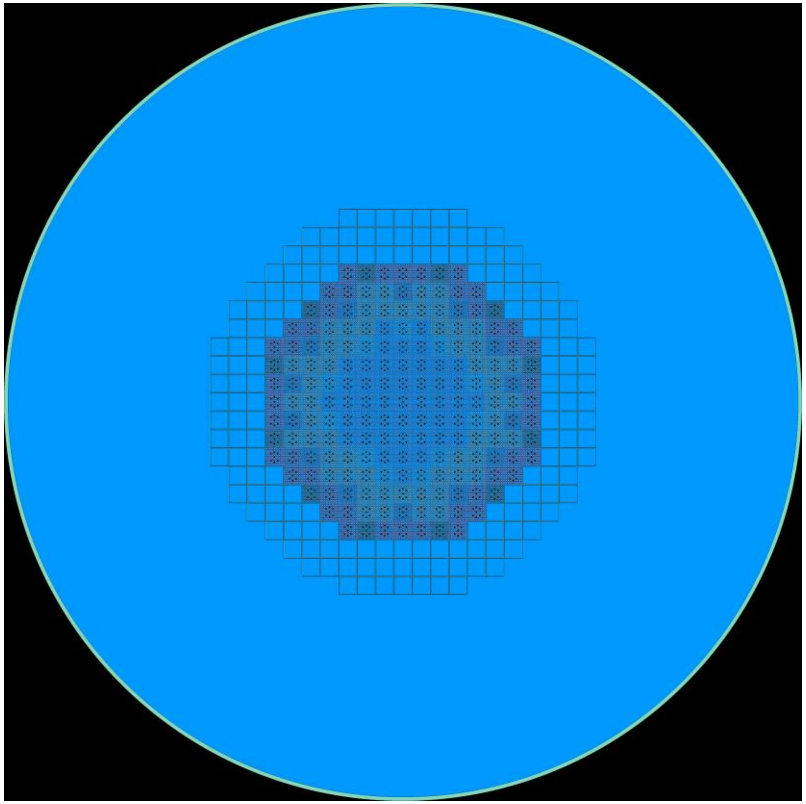
- Sodium-cooled fast reactor
  - Phénix, France
- Lead-cooled fast reactor
  - BREST, Russia
- Accelerator Driven System
  - Myrrha-like (lead-bismuth cooled)
- High-temperature gas-cooled reactor

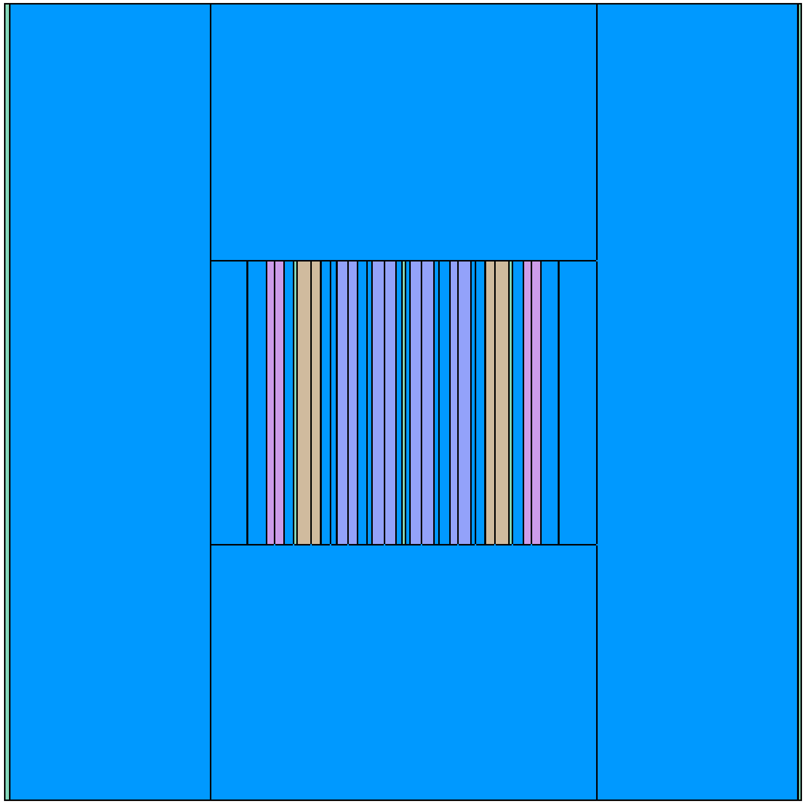
HTGR core



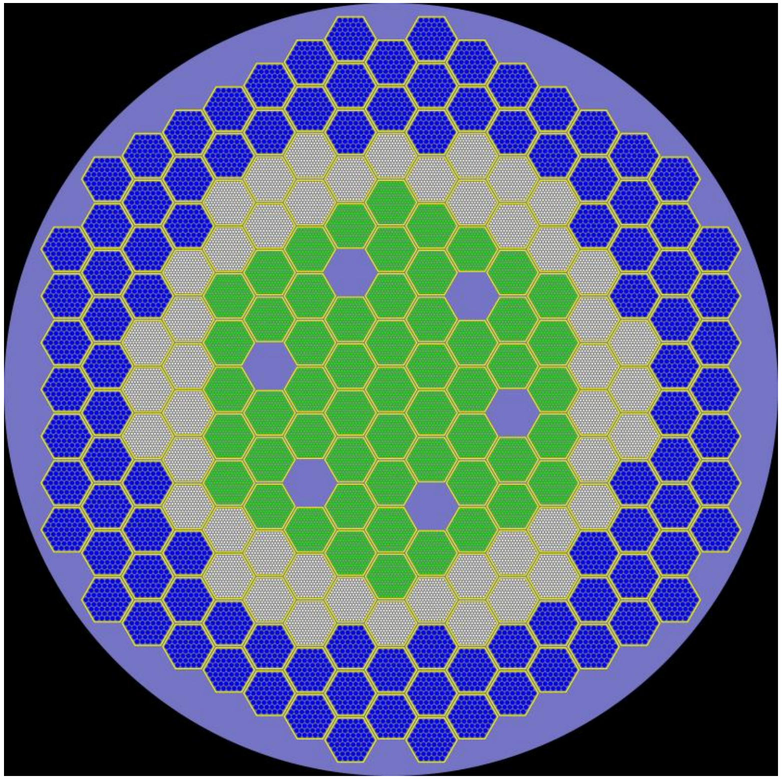


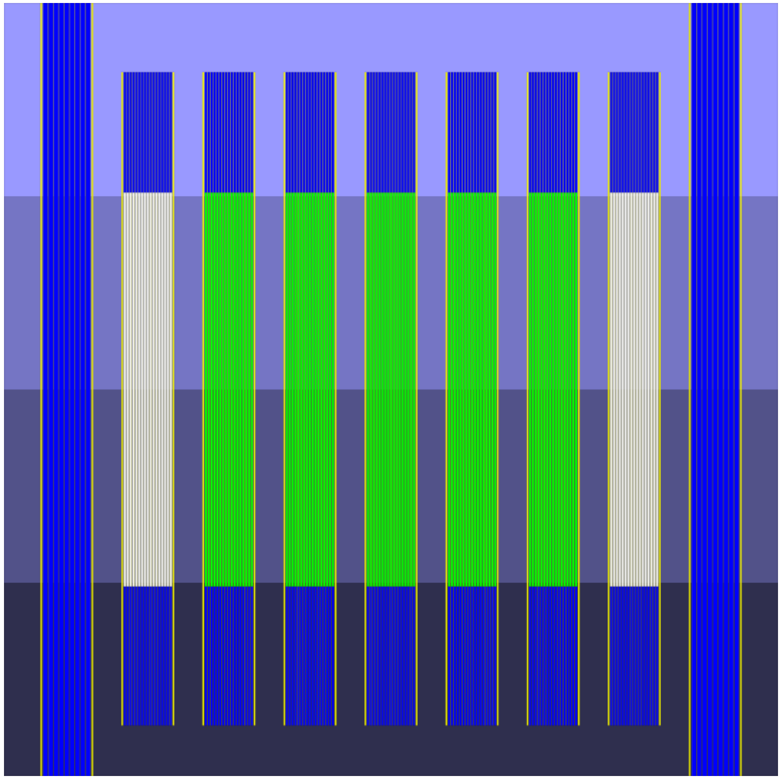
BREST core





Phénix core





Future plans

- New functionalities to the interface
- More reactor libraries

CANDU and RBMK

- Benchmarking
- Economics assessment module

Calculating costs of the entire nuclear fuel cycle with its back-end stage.

Acknowledgements

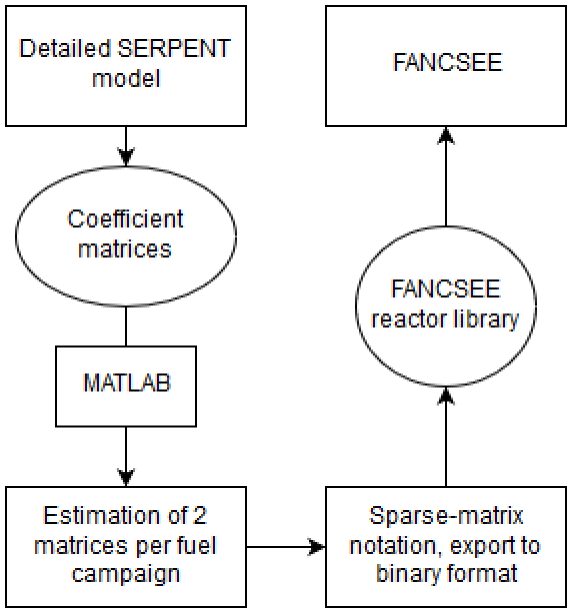
- Part of this project has been funded within the European Project “Brilliant”, Grant Agreement: 662167

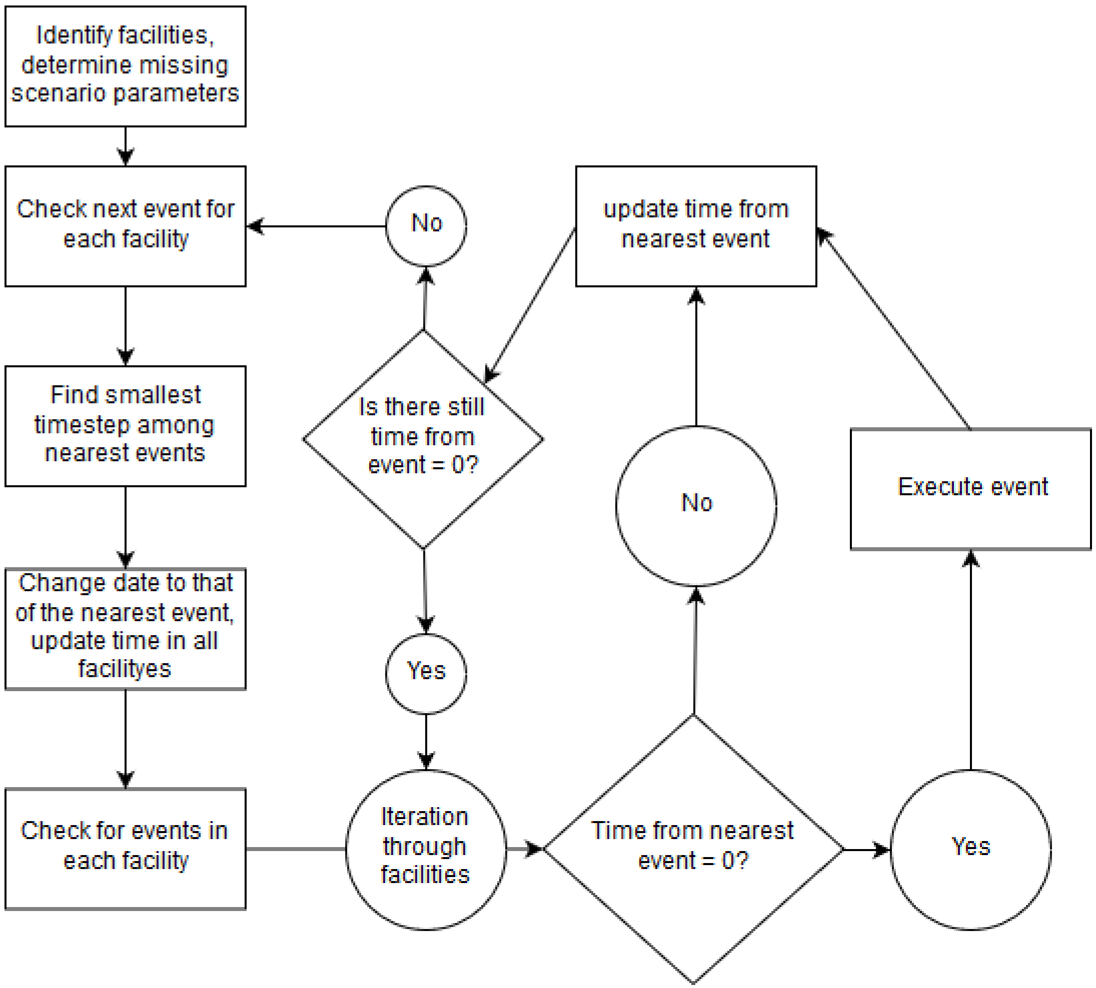
Thank you for your attention!

Questions?

Contact: blazejc@kth.se

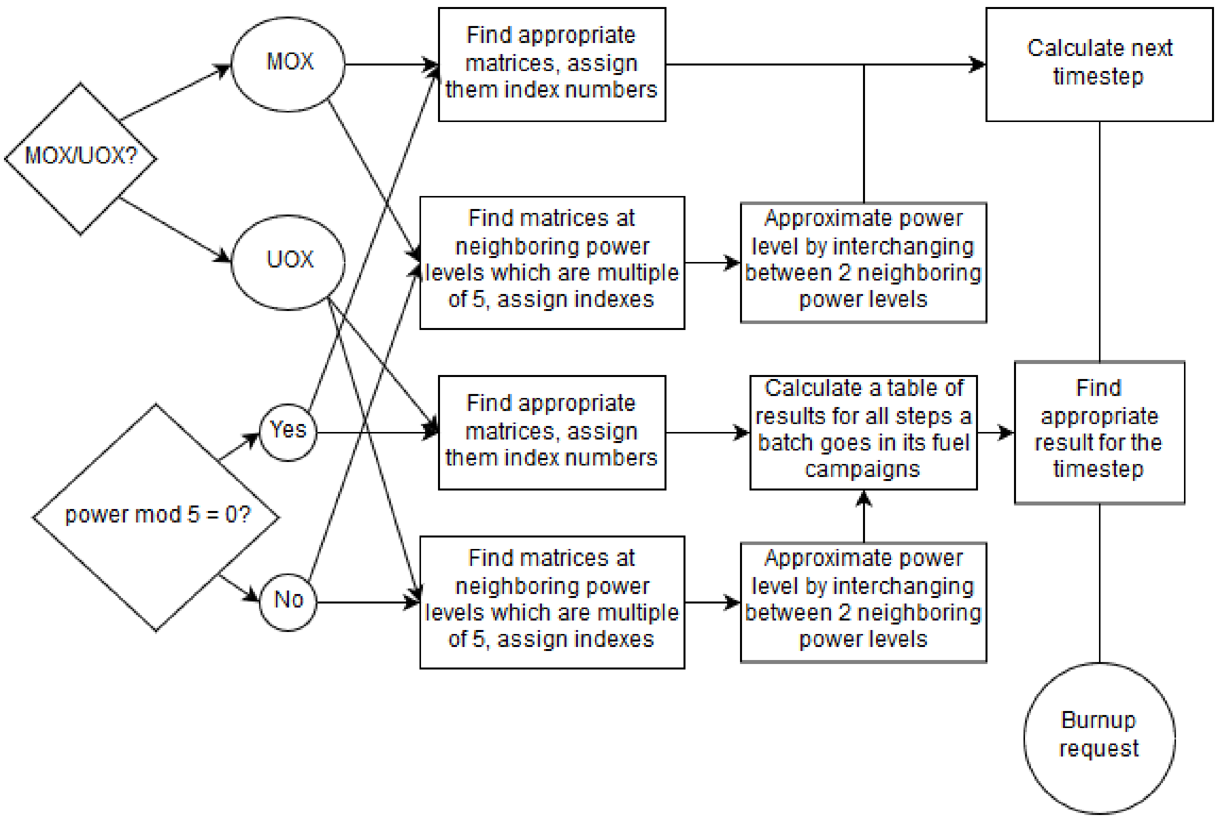
Support slides


Detailed SERPENT model
Coefficient matrices
MATLAB
Estimation of 2 matrices per fuel campaign
Sparse-matrix notation, export to binary format
FANCSEE reactor library
FANCSEE



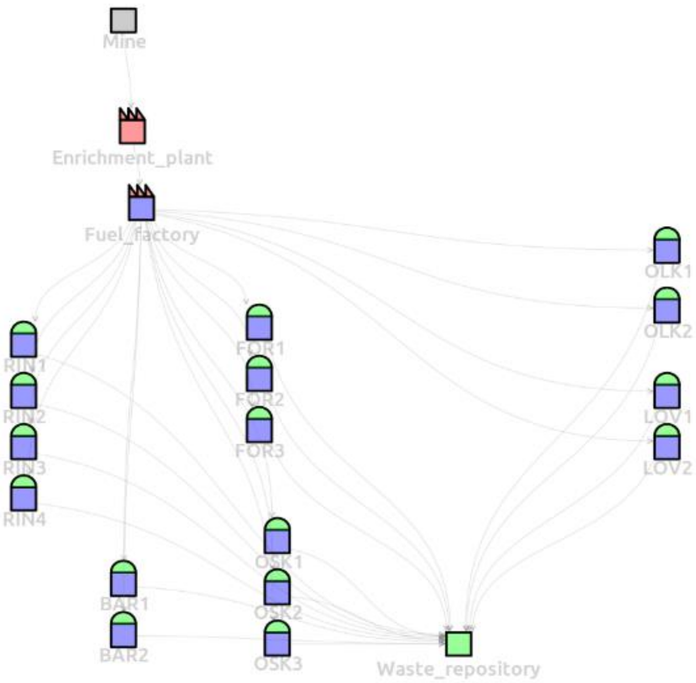
Identify facilities, determine missing scenario parameters
Check next event for each facility
No
update time from nearest event
Find smallest timestep among nearest events
Is there still time from event = 0?
No
Execute event
Change date to that of the nearest event, update time in all facilityes
Yes
Check for events in each facility
Iteration through facilities
Time from nearest event = 0?
Yes

Event	Action	Triggers
Start/stop check	Starts or stops a facility A reactor can only start if all fuel batches are present	Production capacity accumulation (for mines, enrichment plants, fuel factories & reprocessing plants)
Burnup step	Calculate or find next burnup step for all fuel batches in a reactor	
Decay step	Calculate next decay step or find an appropriate solution	
Refuelling step	Stop reactor (stop burnup and power delivery) Move oldest fuel to reprocessing plant / waste repository Check fuel request feasibility, place in delayed fuel requests queue if unsuccessful	Decay step Fuel request Failed fuel request feasibility: - All batches present in the reactor are marked as 'delayed' - Delayed fuel batches cannot use results from a pre-calculated table of results and need to be calculated like MOX
Fuel ready to reprocess	Calculate Pu yield and isolate it, move the remainder to the waste repository	Decay step Delayed fuel request Fuel moved to a waste repository
Fuel too old for reprocessing	Move fuel to a waste repository	Fuel moved to a waste repository
Delayed fuel request	Check fuel request feasibility	If failed – update time from next request feasibility check If successful – fuel request
Report mass	Store atomic density and volume of fuel vectors at the time for each facility	
Fuel request	Move U ore through facilities in the supply chain (produce enriched & DU, move appropriate amount of Pu from a reprocessing plant, create UOX/MOX) until it reaches a reactor Decrease accumulated production capacity in facilities	Reactor start/stop check
Fuel moved to a waste repository	Calculate decay of fuel present in the waste repository prior to movement of fuel and that of the recently moved fuel Merge fuels in the repository into one fuel vector	Decay step


MOX/UOX?
MOX
UOX
power mod 5 = 0?
Yes
No
Find appropriate matrices, assign them index numbers
Find matrices at neighboring power levels which are multiple of 5, assign indexes
Find appropriate matrices, assign them index numbers
Find matrices at neighboring power levels which are multiple of 5, assign indexes
Approximate power level by interchanging between 2 neighboring power levels
Calculate a table of results for all steps a batch goes in its fuel campaigns
Approximate power level by interchanging between 2 neighboring power levels
Calculate next timestep
Find appropriate result for the timestep
Burnup request

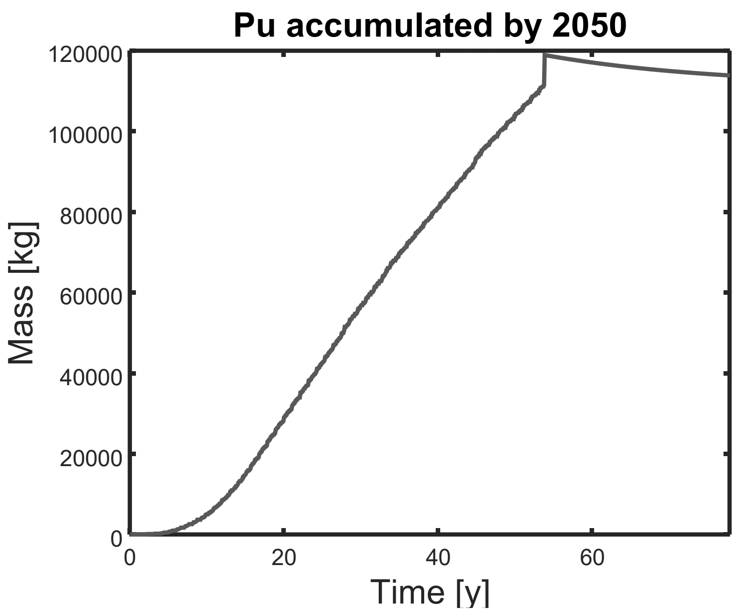
Example scenarios

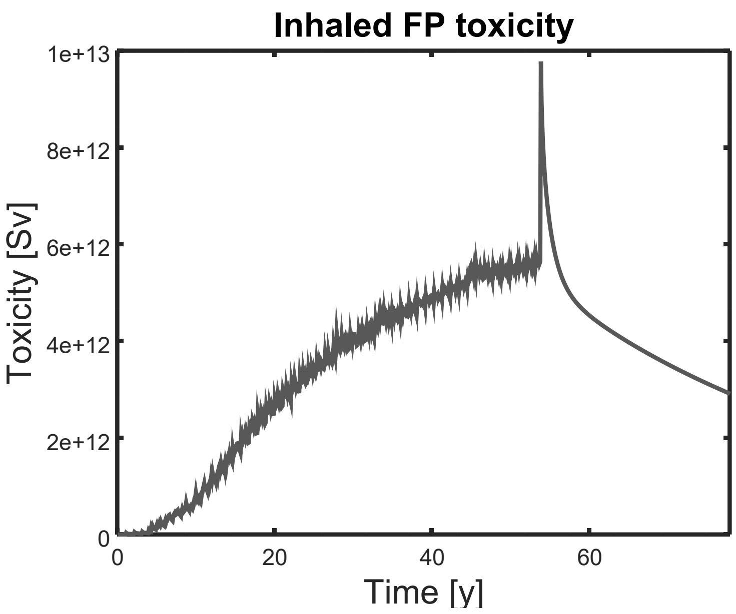
Swedish scenario



Example scenarios

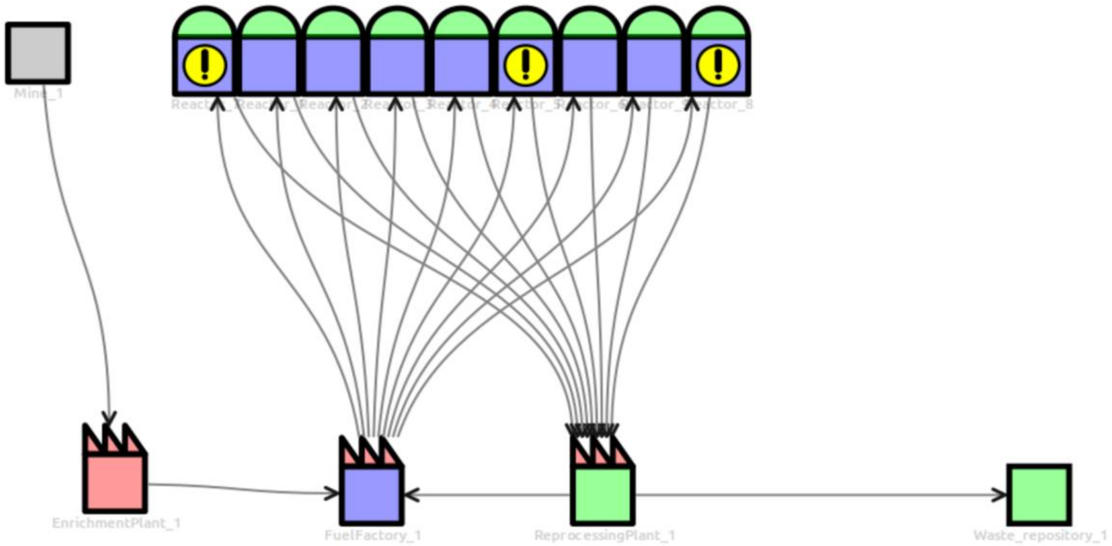
Swedish scenario





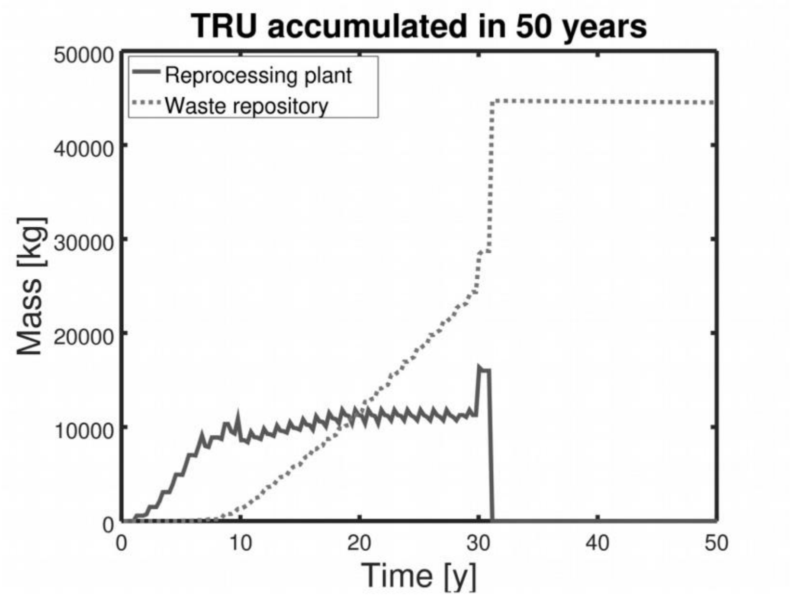
Example scenarios

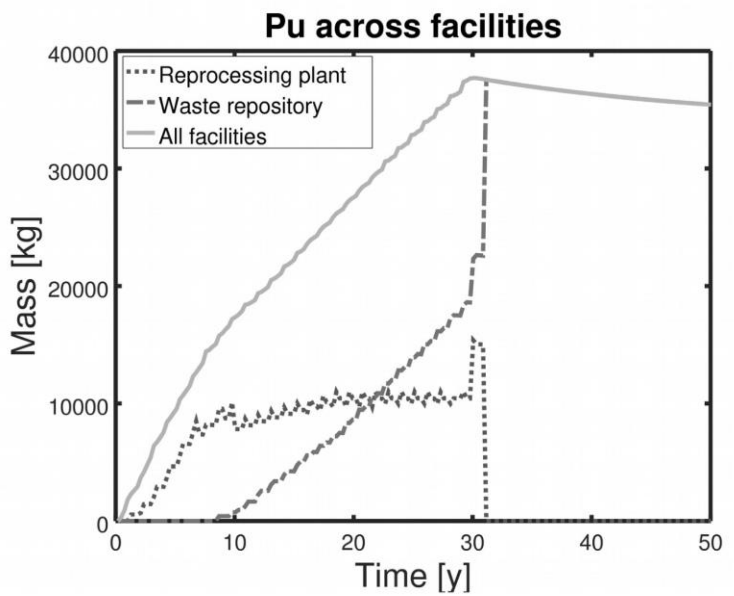
Reprocessing scenario

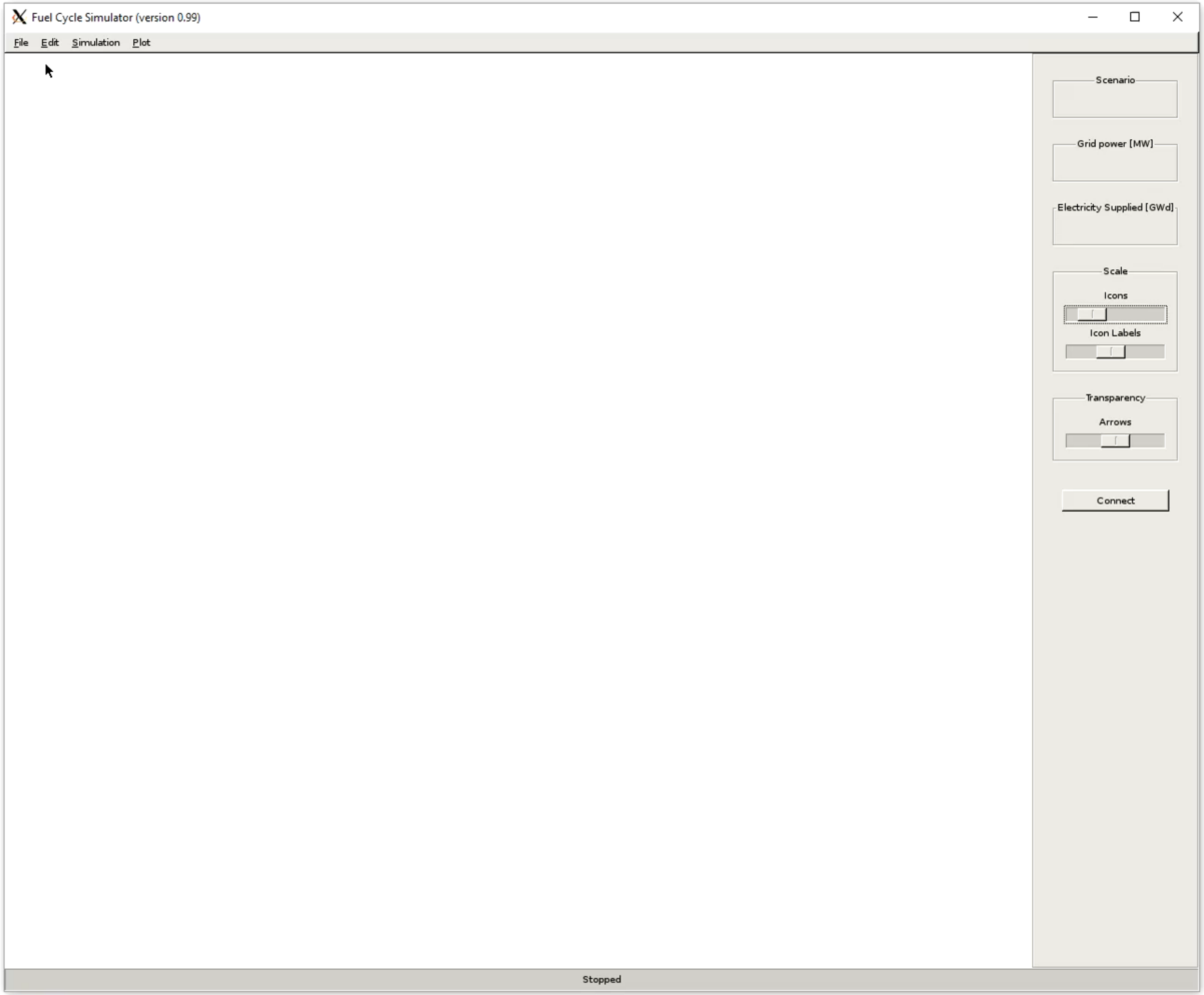


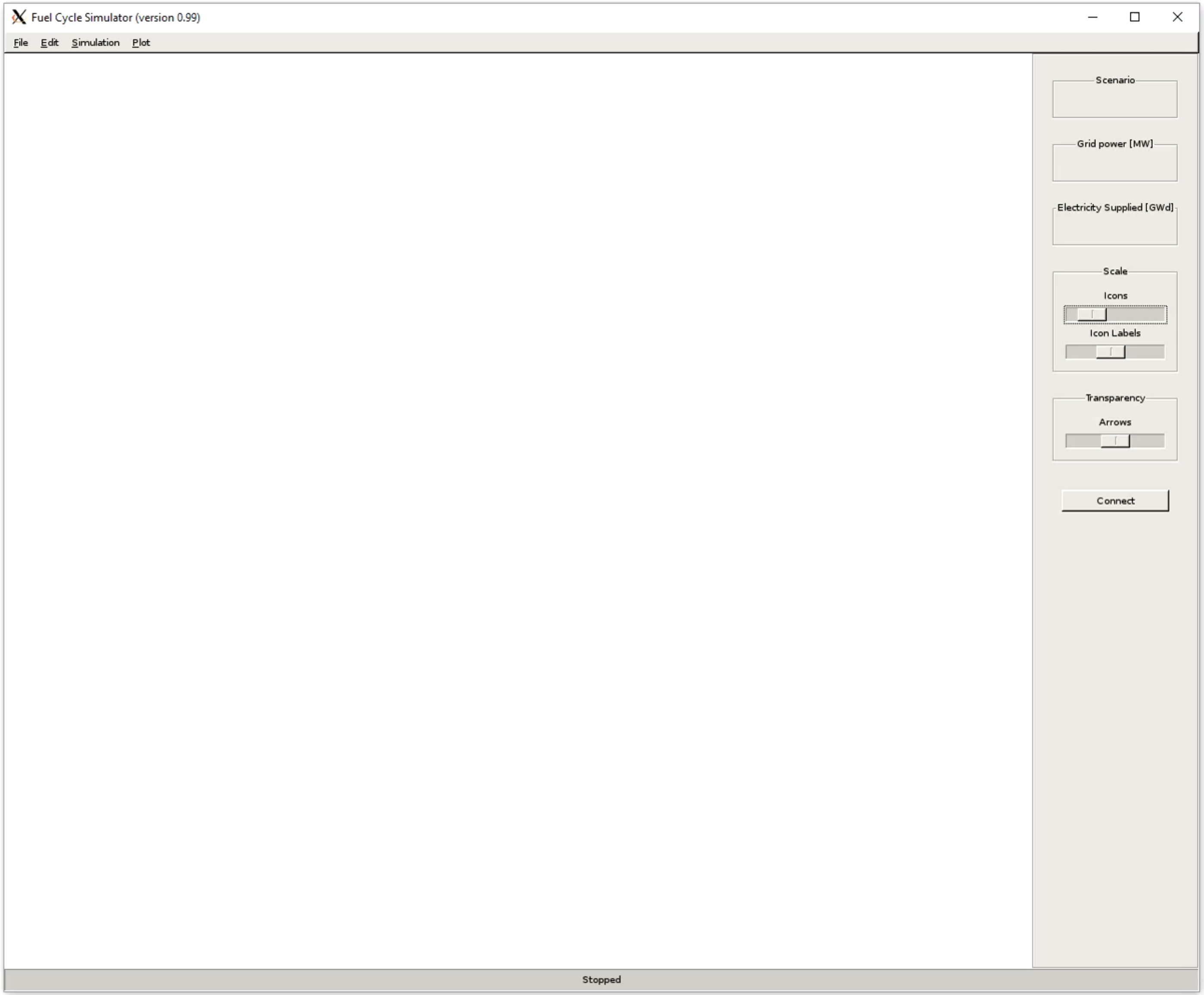
Example scenarios

Reprocessing scenario






Fuel Cycle Simulator (version 0.99)
File
Edit
Simulation
Plot
Scenario
Grid power [MW]
Electricity Supplied [GWd]
Scale
Icons
Icon Labels
Transparency
Arrows
Connect
Stopped


Fuel Cycle Simulator (version 0.99)
File Edit Simulation Plot
Scenario
Grid power [MW]
Electricity Supplied [GWd]
Scale
Icons
Icon Labels
Transparency
Arrows
Connect
Stopped