Nordic nuclear safety research

NKS-388
ISBN 978-87-7893-474-1

Data and visualization solutions for HYBRID core simulation method

William Beere[1]

Tai Tien Huynh[2]

[1]Institute for Energy Technology (IFE), Halden, Norway

[2]Østfold University College (HiØ), Halden, Norway

April 2017

Abstract

The modelling of neutron transport typically relies on two rather opposite approaches: the probabilistic approach, and the deterministic approach. The purpose of the present project is to combine both approaches in order to obtain fast running methods (thanks to the deterministic route) and accurate results (thanks to the probabilistic route). This so-called hybrid method will result in larger amounts of high-fidelity data than previous solutions to this problem.

Viewing, comparing and storing this data should utilize the latest in data handling technology, covering input generation, data storage and output visualization. This report summarizes work performed so far in analysing the data aspects of this problem.

This data system will not only be required to interface correctly with the proposed HYBRID method but will also have to interact with the envisaged user organization. At this stage of the project, the organizations are research institutes and universities. In the future, they may be reactor operators, fuel vendors or even reactor construction companies. Even further in the future spent fuel disposal companies may require some parts of the data solution.

Considering these users we have proposed a list of requirements related to quality assurance, continuous development and aging management. This report makes a start at describing the data problem. Data types, uses and possible database configurations are discussed. Finally, some examples of different data structures are given and possible consequences investigated.

The next project phase will focus on constructing and testing different data solutions and showing possible visualizations.

Key words

Neutron Transport, Database, SQL, NoSQL, Big Data

NKS-388
ISBN 978-87-7893-474-1
Electronic report, April 2017
NKS Secretariat
P.O. Box 49
DK - 4000 Roskilde, Denmark
Phone +45 4677 4041
www.nks.org
e-mail nks@nks.org

Data and visualization solutions for HYBRID core simulation method:

Final Report from the NKS-R HYBRID activity

(Contract: NKS_R_2016_120)

William Beere[1]
Tai Tien Huynh[2]

[1]Institute for Energy Technology (IFE), Halden, Norway
[2]Østfold University College (HiØ), Halden, Norway

Institute for Energy Technology
P.O. Box 40
NO-2027 Kjeller, Norway
Telephone: (+47) 63 80 60 00
Telefax: (+47) 63 81 63 56
www.ife.no

Institute for Energy Technology

Table of Contents

Disclaimer

The views expressed in this document remain the responsibility of the author(s) and do not necessarily reflect those of NKS. In particular, neither NKS nor any other organisation or body supporting NKS activities can be held responsible for the material presented in this report.

1 Introduction

The modelling of neutron transport typically relies on two rather opposite approaches: the probabilistic approach, and the deterministic approach. The probabilistic approach or Monte Carlo approach relies on tracking the individual lives of neutrons, and requires a large computing power for nuclear reactors. The deterministic approach, on the other hand, is based upon fast running algorithms, that solve the problem at hand in only an approximate manner.

The complexity of the problem is reflected in the design of nuclear reactor cores. Figure 1 shows an example of an ABWR. LWR reactor cores are heterogeneous and their inventory develops over time as fuel burns. Both these problems reduce the resulting accuracy of deterministic methods which require simplifying approximations.

The purpose of the present project is to combine both approaches in order to obtain fast running methods (thanks to the deterministic route) and accurate results (thanks to the probabilistic route). The so-called response matrix method was the method investigated in the first phase of the project undertaken in 2016 with NKS support. This method was originally derived in the early seventies in a pure deterministic sense. In the proposed project, the computation of the collision probabilities required for applying the method is carried out using a probabilistic solver instead.

This high-fidelity model will produce a lot of detailed information in space, energy, and angle dimensions. Viewing, comparing and storing this data should utilize the latest in data handling technology, covering input generation, data storage and output visualization. This falls within the domain of Big-data solutions which covers a broad term for data sets so large or complex that traditional data processing applications are inadequate. Emphasis will be placed on the design of the system architecture where flexibility and extensibility will be key factors.

The level of details of the simulations, and the approach allowing a direct computation of whole core problems produces a large-scale data set. There is however, a need to support rapid awareness of the complex 4D (3D + time) data-set for end users. This problem can be divided into;

i) which data is necessary for situational awareness (power, flux, etc.)?

ii) How should this data be visualized for rapid visual perception?

iii) How can the visualization principles be implemented in a software application?

The outcome is enhanced visualization tools. This requires the construction of an adequate data management system with visualization capabilities. In sum, the technology is supporting the efficient development of reactor core simulations, useable first for research purposes by Chalmers, and later by commercial companies.

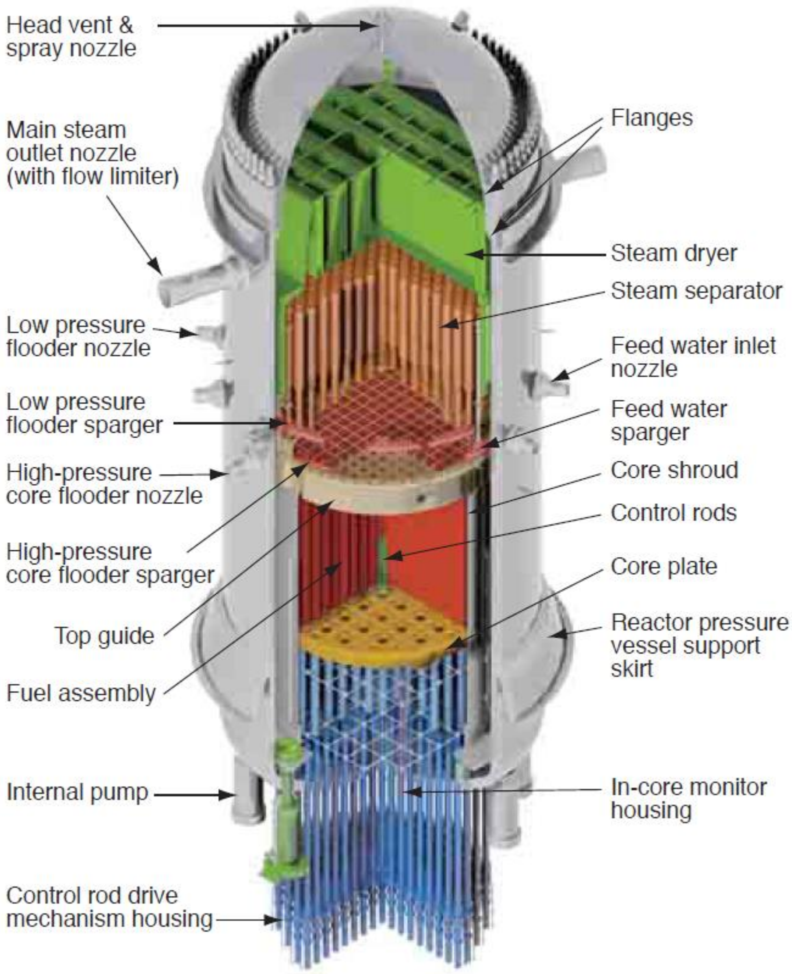


Figure 1. Schematic diagram of an ABWR[1].

There exist commercial solutions to the problem of reactor core management. Examples are ShuffleWorks by Westinghouse[2] and the Studsvik-Scandpower suite of codes [3]. These codes employ graphical interfaces to facilitate fuel management both in the core and during storage.

---

[1] ABWR – Information brochure Hitachi, e.g.
http://www.hitachi-hgne-uk-abwr.co.uk/reactor-pressurevessel.html

[2] Westinghouse X-Works suite. https://x-works.org/

[3] Studsvik Scanadpower Engineering Analysis Services.
http://www.studsvik.com/en/Business-Areas/Nuclear-Fuel-Analysis-Software/

There are also several IAEA safety guides related to this problem [4,5]. Although these discuss requirements and not specific data solutions.

There are also examples of data bases for fuel as in the IAEA Database of measured isotopic concentrations of spent nuclear fuel, with operational histories and design data (Michel-Sendis et al., 2014).

The requirements of these existing solutions mentioned above need to be considered in addition to the needs of method development in this NKS-HYBRID project.

2 Data solution requirements

High-fidelity neutron transport methods as to be developed in this project will produce a lot of detailed information. This information, such as neutron flux, is subdivided into space, neutron energy, and angle of neutron path. In addition, the computation of a whole reactor core would be made possible without the need of pre-computing and tabulating the assembly-wise macroscopic cross-sections as functions of instantaneous and history variables. This requires a complete different approach in data storage compared with current industrial codes, which rely on lattice and subsequent core calculations.

But before data storage options are discussed some basic requirements need to be discussed. Before system optimization can be started suitability to the problem at hand needs to be determined.

The data system will not only be required to correctly interface with the proposed HYBRID method but will also have to interact with the envisaged user organization. At this stage of the project the organizations are research institutes and universities. In the future they may be reactor operators, fuel vendors or even reactor construction companies. Even further in the future some parts of the data solution may be required by spent fuel disposal companies.

Considering these users we propose a list of requirements related to quality assurance, continuous development and aging management. Some of these requirements may come built into database solutions. Others will have to be implemented as an interface to the database. These requirements need to be fulfilled before any performance optimization can be made.

1) Repeatability
Any calculation whose output is saved should be repeatable, this means that the input and the kind of calculation performed also needs to be saved.

2) Traceable
The results of any simulation can be easily traced back to the input and simulation code used to generate them.

---

[4] IAEA Safety Guide No. NS-G-1.12, Design of the Reactor Core for Nuclear Power Plants, 2005.

[5] IAEA Safety Guide No. NS-G-2.5, Core Management and Fuel Handling for Nuclear Power Plants, 2002.

3) Accountable
It should be possible to determine who has constructed a calculation case, or parts of a case. Also a system for checking and approving calculations needs to be supported.

3) Accessible
The data stored can be accessed and processed easily by other codes. This allows for development.

4) Extendible
Adding, or removing, data sections should be not only possible but simple. Simple implies that as few as possible system changes will be required. This should not compromise existing data.

5) Longevity
The data system is expected to be in use for several decades. Underlying computer systems cannot be expected to last for that time frame. This leads to the next requirement.

6) Portability
Data can be transferred from one system to another.

2.1 Data quality

This project will also apply the definitions of information and data quality given below.

1) Authority/verifiability

This relates to who has created the data input and whether the input has been verified by another person.

2) Scope of coverage, Comprehensiveness

In this case this relates to the extent of the problem to be modelled. Either the physical boundaries of the model (edge of moderator, pressure vessel, or containment), or limitations of the problems to be solved (e.g steady state, transients, accident analysis).

The limitations of the scope compared to a complete description give the degree of comprehensiveness.

3) Composition and organisation

Simply this is how the description of the reactor core is to be described as a sequence of data.

4) Integrity

This relates to the correctness of the data set. E.g. that calculation output is correctly related to the input and the calculation code used.

5) Objectivity / Validity

In addition to the accuracy and dependent on the scope the objectivity relates to the limitations on the use of the data.

6) Uniqueness

Simulated, digital data can be easily duplicated; avoiding doing this saves space and time.

7) Timeliness

The results to display from a core simulation will vary depending on the age of the simulation input. Whereas hourly core follow calculations or predictive power change simulations are important during operations, daily, weekly or monthly averages can be more important as time passes.

8) Reproducibility

This means being able to reproduce the same output from the same input. This also implies that the input is available.

3 Nuclear reactor core simulations, information types.

Data for nuclear reactor core simulations can be described in many ways. In this section we discuss some basic data patterns as well as giving some more detailed descriptions.

Many of the data patterns required for the HYBRID project are generic to reactor core simulators. Use should be made of these generic patterns when considering data architecture. However, the actual level of detail in the data required will depend on the intended use and reactor core to be modelled.

On a simple level data can be divided into three categories.

- Input
- Output
- Fixed or Static data

3.1 Basic concepts

3.1.1 Input

As will become apparent this is not a trivial definition. Due to the continuum nature of calculations the output of a previous stage becomes the input of the next stage.

But from the point of view of a calculation the input is the information supplied before the calculation starts. Also, in the context of this assessment the input is defined as the complete set of information required to perform the calculation. This will include the physical nature of the core to be simulated and can also include what kind of calculation to perform.

Input does not include static information which is instead associated with a calculation.

3.1.2 Output

Corresponding to the definition of input the output is the information produced by the calculation. Figure 2 shows an example of output from a core simulation showing axial neutron fast and thermal flux, and power distributions for a single channel. As in this example the structure of the output information may, in part, be identical to the input; the channel and axial divisions are the same for the output as for the input, but in addition the output contains neutron flux in two energies and a value of power. This information can be used to update the burnup information for the relevant fuel. This burnup value is both needed for input and produced by output. It could be considered to simply update this information in a data base, overwriting the input, but this will violate the requirement of repeatability. Thus, maintaining repeatability poses a restriction on the data storage solution with regards to writing new records or updating existing ones.

In contrast to input, output does not have to be complete. I.e. not all results of a calculation need to be saved. Filters or aggregates may be applied to this kind of data. The use of filters and aggregates should to be assessed related to the ease of extensibility of the output information.

E.g. if all output is saved then new filters or aggregates can be easily added. However, if all output is not saved then it may be necessary to repeat the same calculation in order to add new filters or aggregates. This selection is a compromise between the ease (calculation time) of extensibility and the data accumulation rate of new calculations (new output).

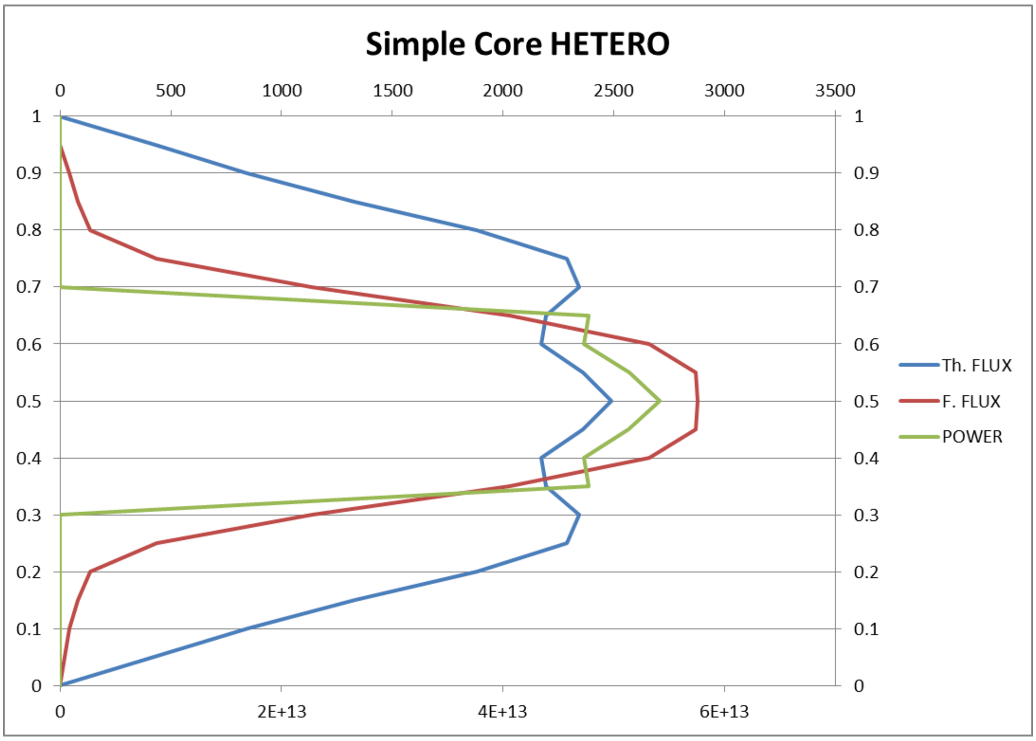


Figure 2. Example of output from a core simulation.

3.1.3 Static information

Static information is similar to input, but it is **identical** for all calculations of the same kind. Exactly which information is static or not depends on the definition of calculation kind.

E.g. for a particular reactor all simulations will contain input information defining the basic shape of the core, e.g. height, number and position of channels. However, for a different reactor this basic information can differ. So if this information is to be classed as static then the same type of calculation (e.g. power distribution) will have to be treated as different calculation types for the different reactors. Alternatively, this basic shape information can be repeated each time as part of the input. In this case the power distribution calculations have the same calculation type for both reactors.

Static information should not be confused with slowly varying input information. I.e. static data associated with simulation which is identical for all executions of the simulation.

Although this appears obvious, when the details of a case are considered it is not so apparent in which category data falls. As an example the basic neutron cross section data will normally be categorized as fixed, but the general core geometry can be either fixed or input. If this is fixed then the simulation code is only useable for a single core geometry. The calculation is then considered hard-coded for a particular core geometry. This inflexibility can have advantages in terms of data requirements.

3.1.4 Calculation

A calculation is any process which takes input information and returns output. In this context the primary calculation is a simulation of a nuclear reactor core. The physics being simulated are at a minimum the fission process where neutron transport is the most important process. Other physical problems can also be considered when simulating reactor cores, such as gamma transport, thermal hydraulics, and heat transfer.

In this context a calculation also refers to the collection of code and *static* information required to perform the simulation. This collection is defined as a calculation kind, shown schematically in Figure 3.

Examples of calculation kinds: 3D Power distribution, burnup accumulation, 2D lattice parameters.

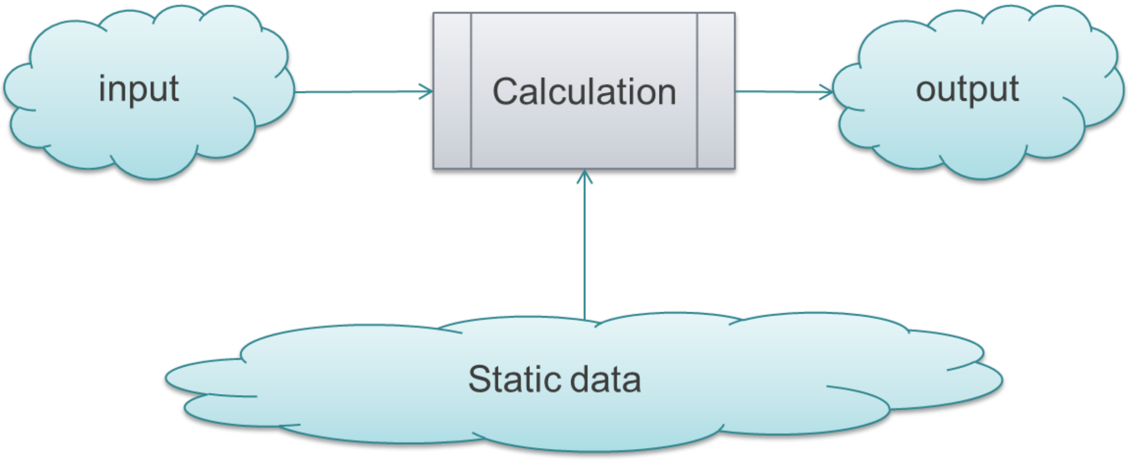


Figure 3. Schematic of relation of input, output and static data to a calculation.

3.1.5 State Calculation

The primary calculation type in this assessment is the state of the reactor core. This is a snap-shot of the condition of the core. i.e neutron flux and power distribution.

For many uses a reactor state calculation input is dependent on the output of a previous state. Visually this can be represented by a chain, see Figure 4. This shows the evolution of different core states (e.g. CL123a) through burnup accumulation (Bu) and refuelling operations (RF). The testing of different scenarios become branches or independent chains.

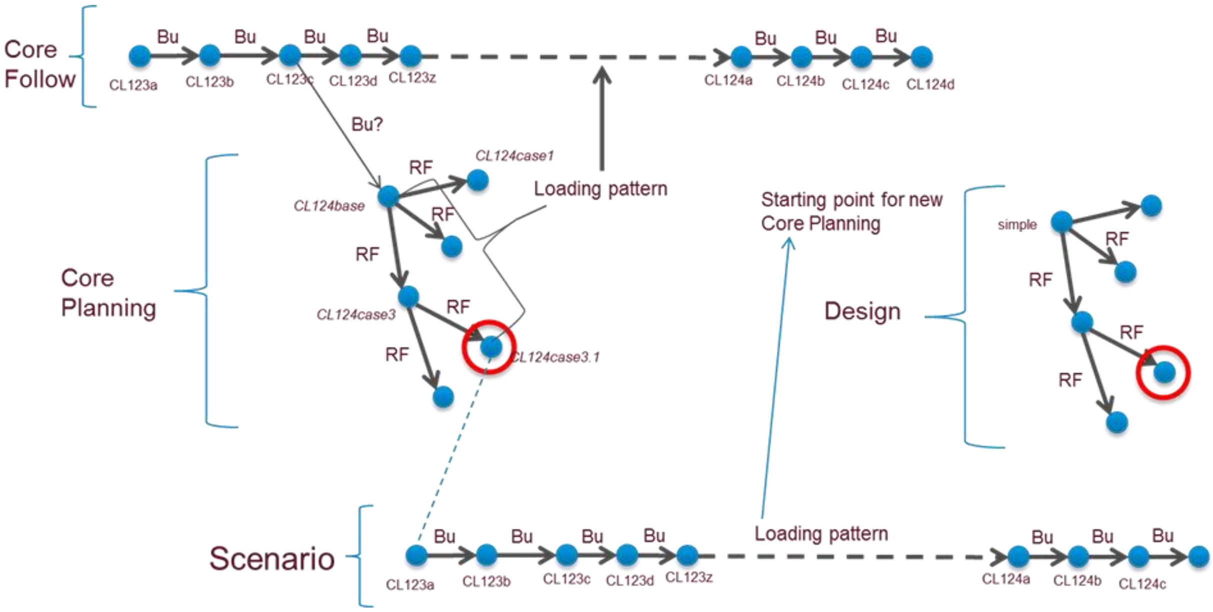


Figure 4. Example of sequential calculations. Input to the next calculation is the accumulation of action of all previous calculations on the original input.

3.2 Physical data elements

3.2.1 Reactor Core

Nuclear reactor power plants contain many processes where the end goal is to supply electricity to a distribution grid. At the heart of the process is the reactor core. This is where the nuclear reaction takes place and energy is generated.

3.2.2 Core Map (loading)

The reactor core comprises of a series of channels. Typically these channels are arranged in a repeating pattern (e.g. square or hexagonal). Each channel has a unique identifier usually indicating its position. Channels may be simply numbered (1,2,3 …) or use a chess board notation (A1,B3 ….), but other variations also occur. Typically a core contains 100-200 channels for a PWR and 600-700 channels for a BWR. Figure 5 shows a quarter core display of an ABWR equilibrium core.

This information is normally static for a particular reactor. Although core modifications can occur, for power reactors they are highly infrequent.

The core map then details which assemblies (defined in the next subsection) are in which channels in the core. This is termed a core-loading. Typically power reactors re-load the core once per year. During this process about 1/3 of assemblies are removed and replaced, the remainder are also re-positioned.

The same geometric pattern is often used for aggregate or meta- data. Such as a map of assembly types, burnup, or power.

Equilibrium Core Loading Map

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
1	3	3	3														
2	3	3	3	3	3	3											
3	3	3	3	3	3	3	3										
4	2	1C	2	2	3	3	3	3	3	3							
5	1C	2	1C	1C	1C	2	2	2	3	3	3						
6	2	1A	2	1C	2	1C	1C	1C	2	3	3	3					
7	1A	2	1A	2	1A	2	1C	2	1C	1C	2	3	3				
8	2	1A	2	1B	2	1C	2	1C	2	1C	1C	3	3	3			
9	1A	2	1B	2	1B	2	1A	2	1A	2	1C	2	3	3			
10	2	1A	2	1A	2	1B	2	1B	2	1A	2	1C	2	3			
11	1A	2	1A	2	1A	2	1A	2	1A	2	1C	1C	3	3	3		
12	2	1B	2	1B	2	1B	2	1B	2	1A	2	1C	2	3	3	3	
13	1B	2	1A	2	1B	2	1A	2	1C	2	1C	2	1C	3	3	3	
14	2	1A	3	2	2	1B	2	1B	2	1C	2	1C	1C	2	3	3	
15	1A	2	1A	2	1C	2	1B	2	1A	2	1A	2	1C	2	3	3	3
16	2	1A	2	1B	2	1B	2	1A	2	1A	2	1C	2	1C	3	3	3
17	3	2	1A	2	1B	2	1B	2	1A	2	1B	2	1C	2	3	3	3

Bundle Type		Enrichment	Number of Bundles
1A	Fresh bundle type A	4.23 wt %	116
1B	Fresh bundle type B	4.22 wt %	76
1C	Fresh bundle type C	4.22 wt %	128
2	One-cycle-burnt fuel		320
3	Two-cycle-burnt fuel		232

Figure 5. Quarter core equilibrium loading pattern for ABWR [6].

3.2.3 Assembly / Channel

An assembly is a joined together collection of fuel and other components. These components form a physical unit to be placed in a core channel. Despite the name channels are not normally separated by any dividing structure. Even in BWR assemblies, see Figure 6, the separating channel box is associated with the assembly and not the core channel.

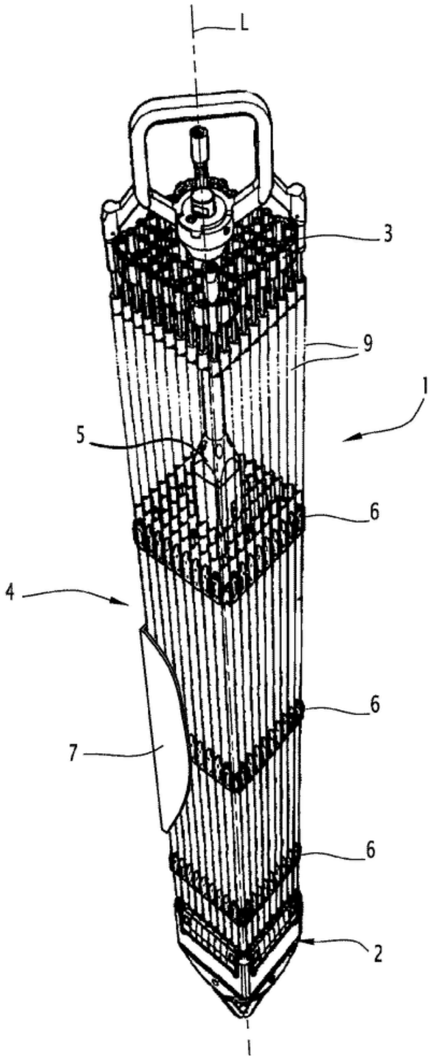
Physically assemblies usually cover the whole height of the core and are either square, hexagonal or round in cross section. The physical shape of an assembly is normally treated as invariant during its life time. However, significant amounts of bowing in an assembly can occur [7]. Many core simulation models do not easily accommodate assembly bowing. However, a high-fidelity code should make provision for such extraordinary alterations to the core model.

Assemblies are unique and always have a unique identifier. This is required as nuclear fuel must be traceable [8].

[6] ABWR Design Control Document, available from the NRC, RS-5146900 Rev.1 Section 4.3, figure 4.3-1.

[7] IAEA-TECDOC-1454, Structural behavior of fuel assemblies for water cooled reactors, 2005

[8] IAEA Safety Series No. 50-SG-QA11, Quality Assurance in the Procurement Design and manufacture of Nuclear Fuel Assemblies, Annex IV, or ISO 10979:1994(en), Identification of fuel assemblies for nuclear power reactors.



1 Fuel assembly
2 Base section
3 Head section
4 Fuel rod bundle
5 Water channel
6 Spacer grids
7 Channel box
9 Fuel rods
L Logitudinal direction

Figure 6. Example of BWR fuel assembly[9].

3.2.4 Assembly type (design / class)

Although assemblies are unique their design is often common. Assemblies are usually all of the same dimensions. A reactor may contain at any time approximately 10 different assembly types. However, during a reactor life time the total number of different types can be much higher (can use 50 as a dimensioning number).

---

[9] BWR NUCLEAR FUEL ASSEMBLY WITH NON-RETAINED PARTIAL LENGTH FUEL RODS, Patent No. US 2012/0243652 A1.

The design includes the number and position of fuel rods and other components. It also includes the type of fuel used. Figure 7 shows a sketch of 4 BWR assemblies surrounding a control blade position. Note that this figure does not give information about the fuel types which can also vary both between and within assemblies.

When including assembly type information it should be noted that assembly design can change during an assembly’s lifetime. One such design change can be the replacement of damaged fuel rods. In this case the design goes from general to unique. In any case the common design only indicates that assemblies have the same starting point.

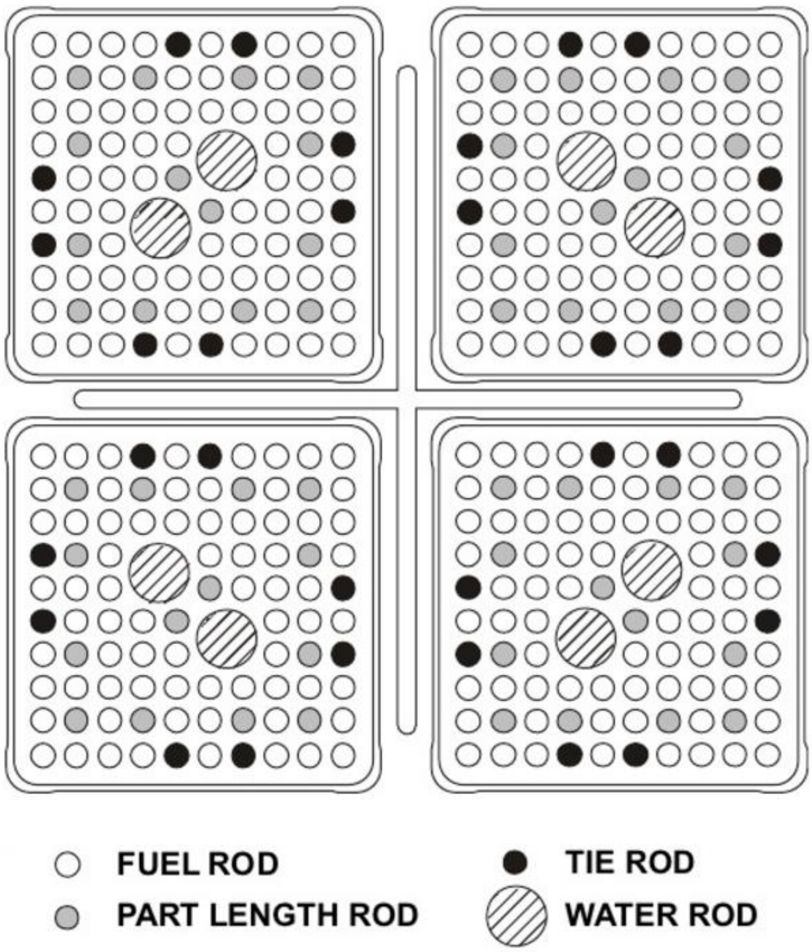


Figure 7. Example of 2x2 GE-14 fuel assembly cross section with control blade (Fennern 2007).

3.2.5 Control rods

All reactors have some form of reactivity control. Control rods are movable neutron absorbing material used to control the total and local power as well as to stop the nuclear reaction in emergency situations. Other reactivity control methods include the use of burnable poisons dissolved in the moderator, such as boric acid.

Some control rods are placed in their own channels in the core, some are placed between normal fuel channels (see control blades in Figure 7) and some are placed inside assemblies. Control rods are always uniquely identified and located in a given position for a core. They can be replaced, moved and associated with different assemblies during core re-loadings.

3.2.6 Node (calculation)

The physical space in the core is often divided into smaller regions for the purpose of calculations. These regions are not physically separate but often follow the same physical construction of the core. E.g. for core channels with a square pattern nodes will also have a corresponding square pattern.

Unlike assemblies and channels which normally cover the whole core height, a node can cover an axial (height in core) range. The general practice is that the same axial divisions are used for all channels. However, this is not a rule.

Nodes are used to identify different axial locations in an assembly / fuel rod for both input and output. Typically, spatial information within nodes is averaged to decrease the calculational cost for modelling an entire reactor core.

3.2.7 Fuel / Fuel rod

A fuel rod (usually cylindrical) is unique and identifiable (requirement of traceability of nuclear fuel). It is usually part of an assembly, placed in a given position. Although it is not impossible for a fuel rod to be moved from one assembly to another.

Fuel has several properties which can be input to some calculations or may be considered as meta-data. These include:

Initial, considered static properties.

- Physical dimensions
- Density
- Material type (pure or mixtures)
    - Isotopic concentrations

Properties as the fuel rod develops

- Temperature
- Power density (typically in W/g)
- Burnup (integrated power, MWd/Tonne [Mega Watt Days per Tonne]
- Isotopic concentrations
    - Some are slowly developing such as Pu inventory
    - Others are quick changing, e.g. Xe (Zenon)

Sometimes these developing properties will be specified as input to a calculation. Sometimes as output. Not all properties are required, this depends on the type of calculation.

3.2.8 Cladding

This is the material surrounding the fuel rod, typicaly zircaloy. Properties of interest are:

Static properties

- Physical dimensions
- Density
- Material type (there are several types of zircaloy)

Variable properties

- Temperature
- Neutron flux (can be several values to distinguish between different neutron energies)
- Neutron fluence (this is the time integrated flux)

3.2.9 Moderator, coolant

The coolant, normally water, is what extracts heat from the fuel rods and in most reactor designs also slows down neutrons (moderates). This slowing down is essential for the nuclear reaction process. There are nevertheless some reactor designs for which the moderator and the coolant can be different media of possible different properties.

As the coolant is a moving medium its properties are variable

- Temperature
- Pressure
- Density (given by relation of pressure and temperature)
    - Void (for boiling the reduction in the amount of water is termed the void fraction)
- Presence of additives
    - E.g. boron (a neutron absorber used for reactivity control) given in ppm (parts per million).

3.2.10 Finer spatial divisions …

The properties of fuel, cladding and moderator can be specified in finer division than their physical limits. E.g. For fuel the power, burnup and possibly isotopic concentrations can differ on a fine scale (say mm by mm) within a fuel rod.

These divisions are artificially added to the simulation model (i.e. they are not physical). They are associated with the type of simulation and their number and description will therefore be fixed for that model.

Different models of the same reactor can have different finer divisions. This affects both input and output.

The number and description of the divisions many follow repeated patterns. E.g. several fuel rods may have the same number and type of divisions.

3.2.11 Aggregated data

As discussed for *output* many values are aggregated. E.g. channel (assembly) power is the sum of all nodal powers for the channel, and in turn nodal power is the sum of all fuel rod powers in the node. Also fuel rod power is the sum of powers in the finer divisions of the fuel rods.

3.2.12 Other historical state values

Spatial aggregates are not the only types. Many aggregates are in time such as fuel burnup. There are also other aggregate types of interest such as:

Historical boron – time averaged boron concentrations in the moderator.

Historical void/density – time averaged void (for boiling water) or density of the moderator.

Historical control rod use – time averaged presence of control rods in or nearby the fuel assembly.

These historical values are usually time averaged over the individual assemblies (assembly node) lifetime. They are important in traditional simulations which use assembly property libraries generated from 2D lattice codes.

4 Simulation kinds

The use of the system will vary during its lifetime and may mean that no single data-base architectures will be optimal. The proposed HYBRID system is in a developmental stage. Here the same simulation input will be tested against different calculation codes. During the testing stage many different independent simulation inputs (e.g. benchmark cases) could be tested against one or few calculation codes. Later as the system matures to delivery state simulation inputs are no longer independent but linked to the output of a previous simulation. For the delivery stage, envisaged uses could be:

4.1 Core Follow

The simulation mimics the actual reactor core. Measured data such as moderator pressure, temperature and total reactor power are used as input. The resulting output of power distributions are used to update burnup / isotopic concentrations at the desired level of aggregation.

4.2 Core Planning (re-loading)

Alternative core loadings are simulated, often including several future core re-loadings. Performance over several cycles can then be compared.

- Safety assessments
- Cycle assessments

4.3 Power change planning

The core behaviour over short term power transients can be predicted to ensure / optimize performance against operating criteria.

4.4 Code development testing

In this case keeping track of not only the input and output but also which code(s) have been used is essential to be able to properly compare results. Both models and codes are continuously changing so the system needs to easily accommodate changes.

5 Data bases

To date there is considerably more choice of data storage systems than previously available. Options range from the use of file naming conventions, relational databases (such as SQL type) and document based databases (such as noSQL).

The requirements outlined in section 2 form part of the acceptance criteria for database selection. In addition optimization metrics related to performance should also be developed. Such metrics can be retrieval time of data and accumulation rate of data. These parameters need to be assessed related to the intended use of the system and the detail of data to be stored.

5.1 Database management systems (DBMSs):

When considering database choice the amount of development support and possible longevity of a database solution should be assessed. Popularity of database solutions can be considered a good metric for this assessment.

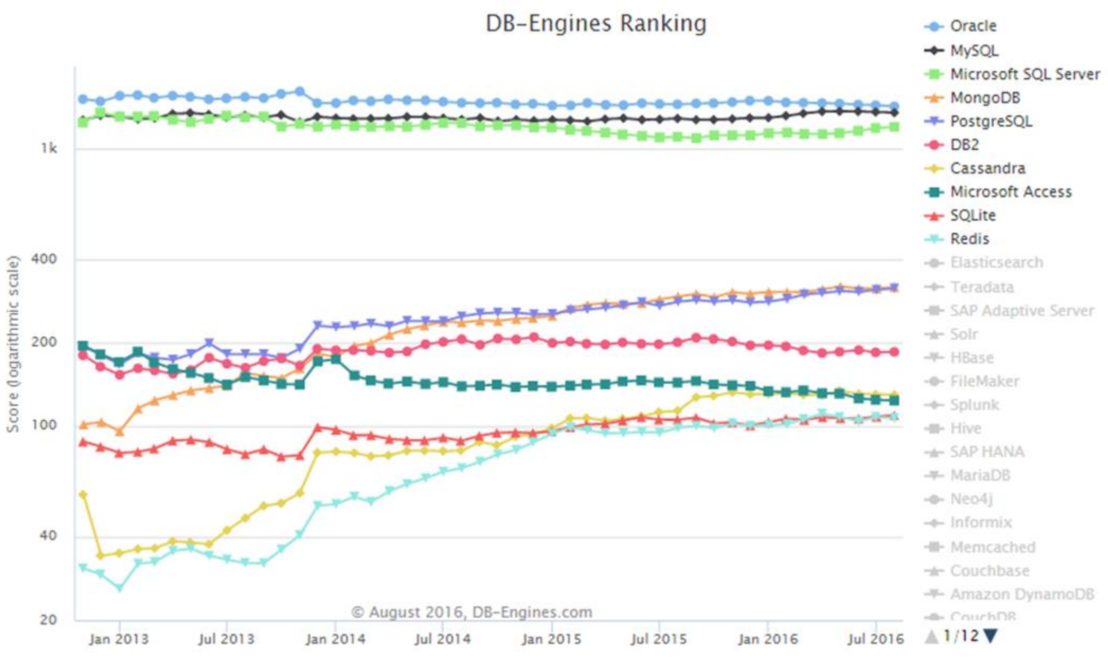


Figure 8. DB-engines ranking.

Figure 8 represents the top 10 DB engines in August 2016 [10]0. This figure shows the popularity of different databases based on the following criteria:

- Frequency of technical discussions about the system.

[10] “DB-Engines Ranking - popularity ranking of database management systems.” Available: http://db-engines.com/en/ranking. [Accessed: 15-Aug-2016].

- Number of mentions of the system on websites.

- Number of profiles or job offers where the system is mentioned in professional networks.

The top three data bases (Oracle, MySQL and Microsoft SQL) are all table based relation databases. They are all share a common interface language (SQL). Some of the other databases which are increasing in popularity are MongoDB, Cassandra and Redis are NoSQL (document based) databases.

5.2 Relational database management systems (RDBMSs):

RDBMSs are based on the relational model as invented by E.F. Codd. This is popular and common choice for the information storage since 1980s. In RDBMSs, we need a predefined schema and systems organize data in table-based structure. Even though NoSQL databases have been developing rapidly and are being used more widely, it is no coincidence that relational DBMSs still dominate the database engines ranking (Figure 8). The top three DBMSs scored about four times more compared to the fourth candidate – MongoDB (NoSQL).

RDBMSs have high transactional based support with an emphasis on ACID properties (atomicity, consistency, isolation and durability). This is both an advantage and also a disadvantage for such databases. The database performance becomes adversely affected if we have many connections with transactions to edit the data at the same time. In the case of core simulations this is not expected to be a problem. Most operations are either read or write with few updates. Updates are expected to conflict with requirements of reproducibility. Furthermore, if we follow the user-context, the delete operations would also be infrequent and only associated with database clean-up where speed is probably not a critical factor. In addition, it is probably that only one or few users will be involved in writing new data to the system at one period of time. Therefore, it is not necessary to choose RDBMSs to acquire these advantages. Besides, for core simulation data the need to allow evolution of the data structures stored puts tables based relational databases at a disadvantage.

5.3 Not only SQL database management systems (NoSQL):

Not all database applications have pre-defined structure and design from the start of a project (or commissioning of a data store). For core simulations the data structure is expected to differ between different uses. This suggests that support for unstructured data sets will be required (schema less). It is also observed that many applications which require big data, real-time web and the appearance of web 2.0 (Facebook, Google and Amazon) use the new generation of databases – NoSQL.

NoSQL brings advantages when the first priority is flexibility of data structure, it leads to problems with data integrity or data constrains. This requires that database developers ensure these requirements. Also, such databases have difficulties in handling many to many relations.

There are 4 categories of NoSQL databases:

1. Key-values stores organize data as a hash table with a unique key and a related value.
2. Column stores have a better structure than key-value stores when each value is formed to multiple columns.
3. Document-based where documents can be JSON documents with a well-organized structure.
4. Graph-based NoSQL arranges data with graph theory, vertices and edges. This treats very well the relation between objects, entities and algorithms to traverse through them.

In the context of this project, the key-values and column stores are not suitable with a simple data organization. The document-based NoSQL seems to be the most reasonable while the graph one can be applied as well, especially when we need to process algorithms to query data along a core follow or core planning sequence of states.

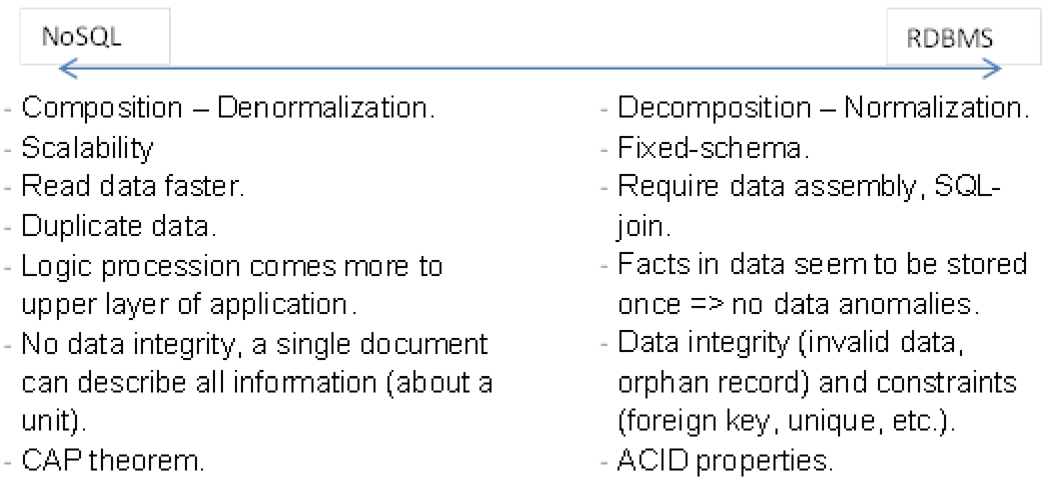


Figure 9. Comparison of NoSQL to RDBMs.

Figure 9 shows us the advantages and disadvantages when useing NoSQL or RDBMS. The more scalable our system is, the lower the data integrity level will be. Analysis of the data structure for our system will give us more information about the relation and application between these two types of databases to our system.

5.4 Entity Relational Model (ERD) of a generic core simulator:

There are many generic aspects of reactor core simulation data. For example the core is physically organized in channels which contain fuel assemblies of a few generic types. This descriptive view of data types can be used to determine an entity relational description (ERD). An example of an ERD is shown in Figure 10 indicating how entities are linked in a form of natural language. In this figure the different objects, state (calculation), node, assembly, etc. are linked by their relation as by the Chen-notation. These links are either 1 or n indicating if the relation is to one or to many. In this figure many States have many nodes where the nodal burnup (NodeBU) can be used to link the correct node to the correct state. Each node is associated with a single nodal information table (NodeInfo) of which there are many.

This describes the entities relation for the system. This kind of representation can be used to describe the interconnection between different types of data. This structure can be determined regardless of whether the data is then stored in tables or documents.

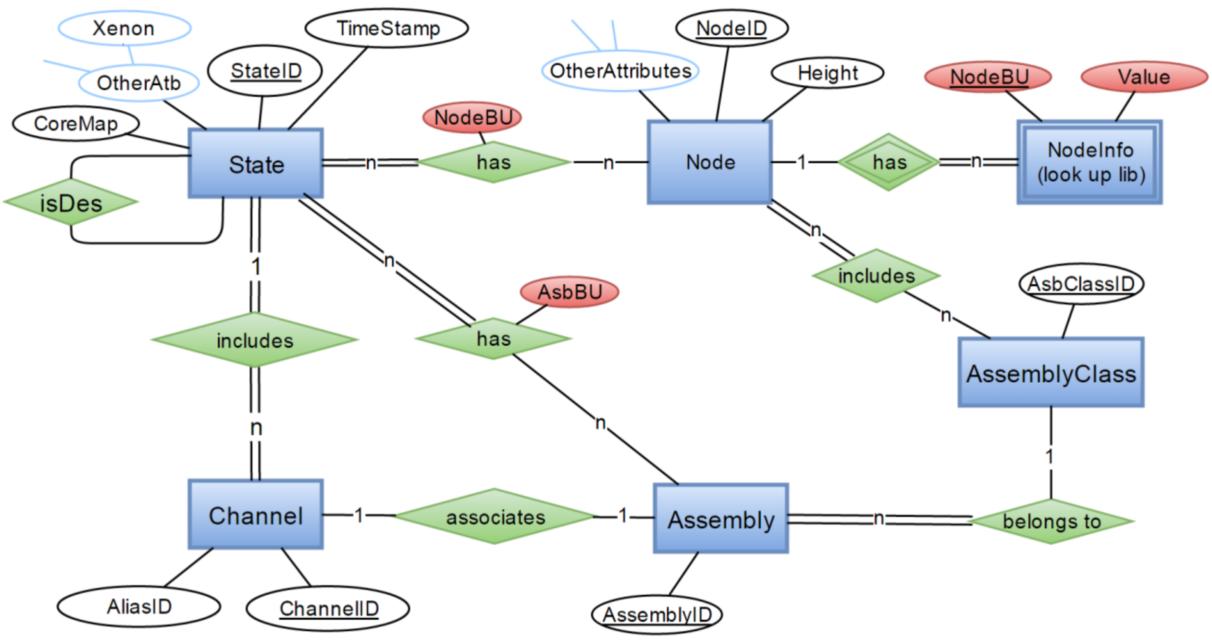


Figure 10. ERD of a generic system

6 Table or document data format.

In this section the concepts described above are used to assess individual elements of proposed systems. Considering a simplified calculation input consisting of a core map describing a loading pattern where the assembly burnup represents the historical parameter to describe the current state. Then the assembly power is the output value of interest.

We can then define the calculation type of core follow as being an operation to return the new assembly burnup values given the current power level and time increment. Output is also defined as the new power levels.

The core planning operation similarly requires a new core map and outputs the corresponding burnup map and initial power levels (definitions of fresh fuel assemblies will also be required).

6.1 Alternative representations of a core-map.

As discussed above each data structure can be stored and displayed in many different ways. This section gives some examples for data storage of the core map, consisting of 2D information on assembly type, ID (unique identification number) and assembly burnup. Use of a single burnup value for an assembly is a simplification and chosen here for illustration purposes.

From the point of view of running a simulation of the core the simplest view is of three maps showing the different data types, see Figure 11.

ID	**1**	**2**	**3**
1	AA0001	AA0101	AA0004
2	AA0102	AA0003	AA0104
3	AA0002	AA0103	-

Type	***1***	**2**	**3**
1	A	B	A
2	B	A	B
3	A	B	-

Burnup	**1**	**2**	**3**
1	10	0	12
2	0	0	20
3	12	20	-

Figure 11. Illustration of core-map data in as 2D data maps

This information format is also suitable for a simulation code based on representing the core as a regular square lattice. However it offers little flexibility for data storage in this format as it relies on fixed array structures.

The next alternative is to use a table to represent the core loading pattern and associated information, see Figure 12. The fact that fuel assemblies carry a unique identification number can be utilized. This should be considered mandatory for core-follow calculations. However for core-planning calculations it is very likely that calculations with fictitious assemblies would be performed. The same is true for code development and testing. The condition of uniqueness for assemblies in such test calculations is a theme which will be addressed in the next phase of this project.

X	Y	ID	Type	Burnup
1	1	AA0001	A	10
1	2	AA0101	B	0
1	3	AA0004	A	12
2	1	AA0102	B	0
2	2	AA0003	A	0
2	3	AA0104	B	20
3	1	AA0002	A	12
3	2	AA0103	B	20

Figure 12. Illustration of core-map data as a table.

The advantage of this view is the reduction to a single table. However it is harder to see the relation of the assemblies to each other. Additional information can be added as extra columns or the table can be split into two or more tables with rows identified by the unique assembly ID. For this simple example this solution works well.

When considering more detailed core descriptions with multiple values describing assembly exposure at multiple axial levels and even down to fuel rod level then adding columns or tables increases the amount of excess information required. This can be illustrated by considering axial burnup, see for example Figure 13. Here the assembly node is repeated many times.

ID	Node	Burnup
AA0001	1	9
AA0001	2	10
AA0001	3	12
AA0001	4	10
AA0001	5	9

Figure 13. Adding axial burnup as additional table to core-map data.

When this structure needs to be repeated in a data-base for multiple calculations then additional information is required to identify which calculation the information relates to. This increases the amount of storage space required and invariably requires additional tables to carry the information about the calculation type.

In general tables are very good at representing simple data structures but offer little flexibility and extensibility. Alternatives are to use collection based storage structures such as XML or JSON. Although there are definitive advantages to JSON (Nurseitov, et al. 2009), XML is still the more commonly used of the two. Figure 14 shows the core map information presented as XML.

```
<Assembly ID=”AA0001”>
	<Channel X=”1” Y=”1”/>
	<Type>A</Type>
	<Burnup>10.0</Burnup>
</Assembly>
<Assembly ID=”AA0101”>
	<Channel X=”1” Y=”2”/>
	<Type>B</Type>
	<Burnup>0.0</Burnup>
</Assembly>
<Assembly ID=”AA0004”>
	<Channel X=”1” Y=”3”/>
	<Type>A</Type>
	<Burnup>12.0</Burnup>
</Assembly>
...
```

Figure 14. Core-map data as XML document.

This information can also be split into different parts. One option would be to change the channel description to an ID (e.g. 1-1, 1-2, 1-3 …) and have another collection indicating the physical position of each channel. This splitting of information has the disadvantage that information needs to be pooled in order to get a full picture, but it has the advantage that the description of the assembly is more generic and less dependent on the reactor type. It is also possible to change the hierarchy listing first channels and then describing which assembly is to be placed in them.

One of the tasks of the continuation of this project will be to assess these different options related to the criteria given in chapter 2, the uses given in chapter 4 and the analysis examples given in section 6.2.

To demonstrate one of the advantages of this type of information structure the inclusion of axial burnup distribution could be implemented as shown in Figure 15.

```
<Assembly ID="AA0001">
     <Channel X="1" Y="1"/>
     <Type>A</Type>
     <Burnup>10.0</Burnup>
     <BurnupProfile>
          <Node Z="1">9</Node>
          <Node Z="2">10</Node>
          <Node Z="3">12</Node>
          <Node Z="4">10</Node>
          <Node Z="5">9</Node>
     </BurnupProfile>
</Assembly>
...
```

Figure 15. Core-map data as XML document with addition of axial node burnups.

It would be possible to continue to build up the entire core in a single XML document and even to group collections of successive core calculations in the same document. However at some point this document becomes unreasonably large and will result in slower operations to extract, change and add data. One solution to this problem is to store XML documents in a document database as discussed in section 5.3.

The challenge is then how to split this information to achieve best performance. To assess performance example uses of the data need to be developed. In chapter 4 different kinds of simulation were described, these form some test cases. In addition different queries to extract data from multiple calculations need to be added to the use cases.

6.2 Data query example

Extending the examples given above a data solution in table structure is visualized in Figure 16. This shows the core-map, burnup and power tables and how sections of them need to be gathered to form calculation input. The placement of new output data is also shown. There new core-map data is only required when there is a core planning operation.

Alternatively the data can be organized in a document structure, see Figure 17. There it can be seen that a document contains all core-map, burnup and power tables. This forms the entire input and output from the core follow calculation. The core planning calculation requires an additional piece of data to specify changes to the core-map.

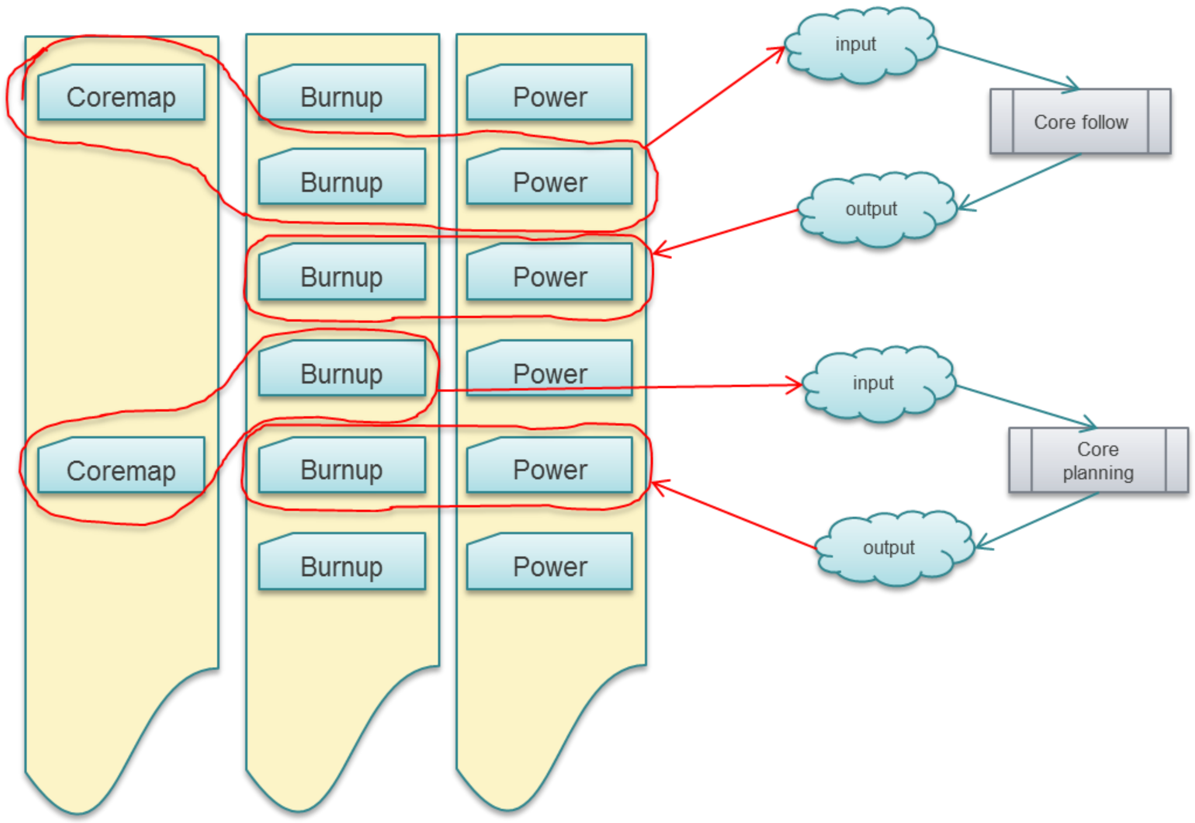


Figure 16. Simple simulation model with data in table structure.

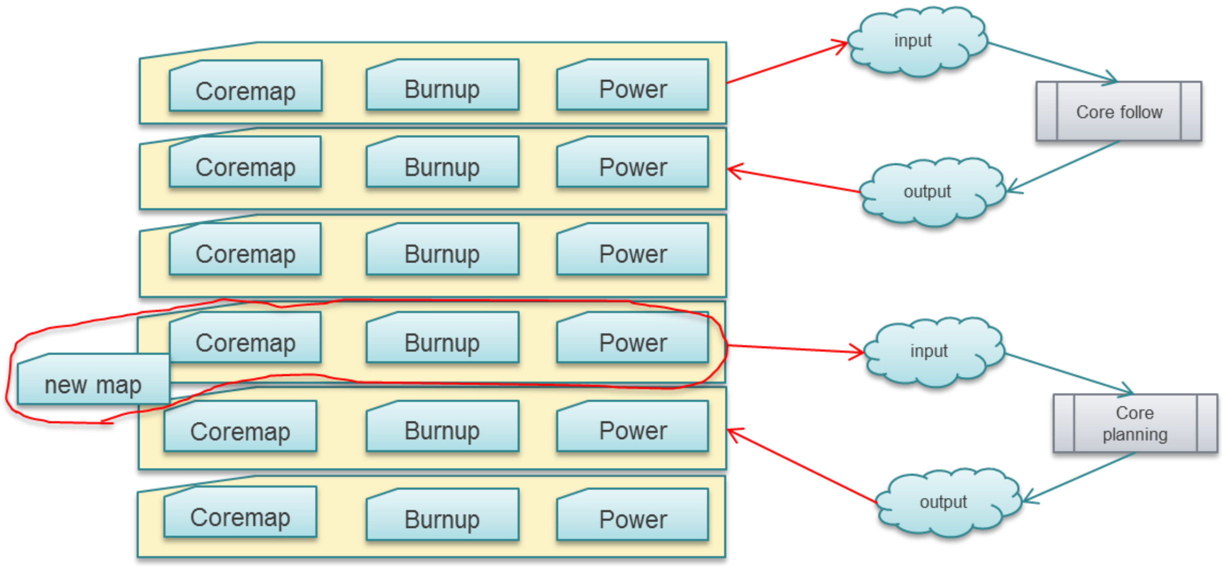


Figure 17. Simple simulation model with data in document structure.

Comparing these two alternatives it is seen that the core-map data is repeated in the document data solution. This will increase storage usage but may give other performance enhances such as speed of access or robustness of data.

As an example consider an alternative data query to extract information on a single assembly along a sequence of calculations. Figure 18 and Figure 19 visualize such a query for the two alternative data structures. The green line indicates where the core-map provides the index information to locate both burnup and power for the queried assembly. Note that the index shifts after a core planning operation.

Comparing these alternatives the index is determined fewer times with the table based data than with the document based data. However, in actual database solutions index tables can be set up to speed up such operations. Solutions can vary as to whether indexing is managed by database management system or provided as part of the internal database support functions.

Such cases need to be constructed to test different database solutions. Such cases should include the effects update and delete operations have on the ability to retrieve and analyse data.

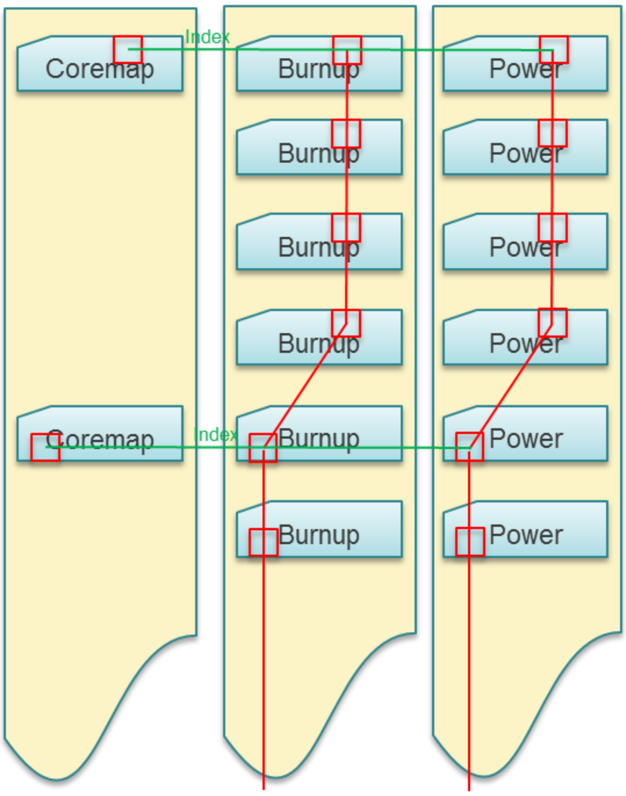


Figure 18. Example of assembly data trend query for tabular data structure.

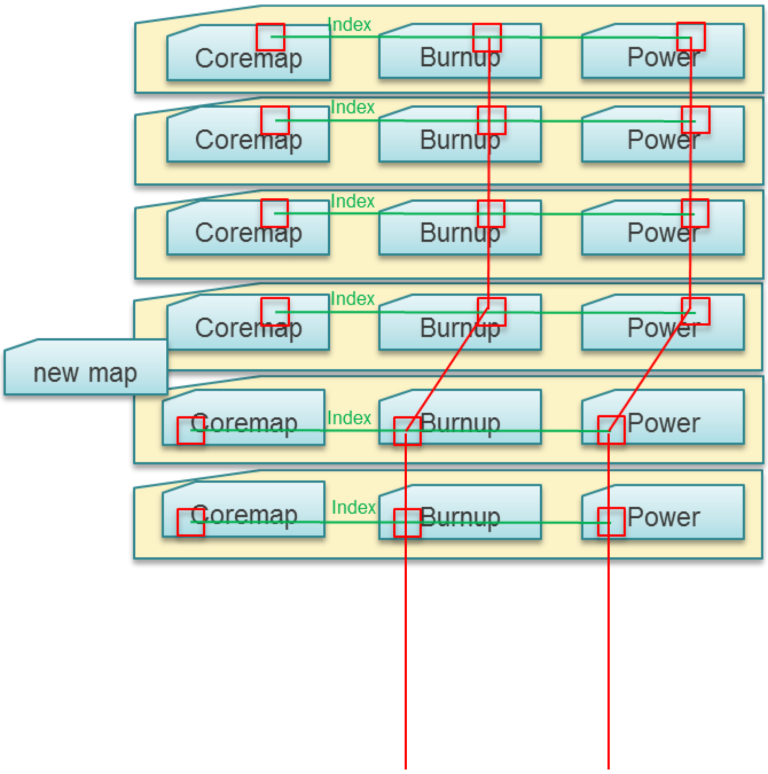


Figure 19. Example of assembly data trend query for document data structure.

7 User interface, data visualization

The user interface (UI) can be different from reactors to reactors, but it can still have some similarity. There are also different levels of detail, from the overall system perspective showing the linking of different state calculations to detailed pin wise power distributions for individual assemblies.

Each type of view may require a customized UI. However, there may be advantages to maintain some form of familiarity between visualizations at different levels of detail. There are also benefits to system design if elements can be reused. This can reduce the development time required, especially as the system evolves.

Generic visualizations capable to showing the underlying data structures should also be considered. This will increase the accessibility of the data.

Initial work in this project so far has been focused on data structures. However, how and which structures are to be visualized is equally important. The kind of data required for visualization will have a direct impact on the choice of the underlying data structure.

For example for core planning operations a core-map like visualization would be the most appropriate, see example in Figure 20. This is a 2D data representation where only assembly averaged values may be relevant. Alternatively core follow operations may require a more detailed view, see example in Figure 21. This is a representation of 3D nodal neutron flux.

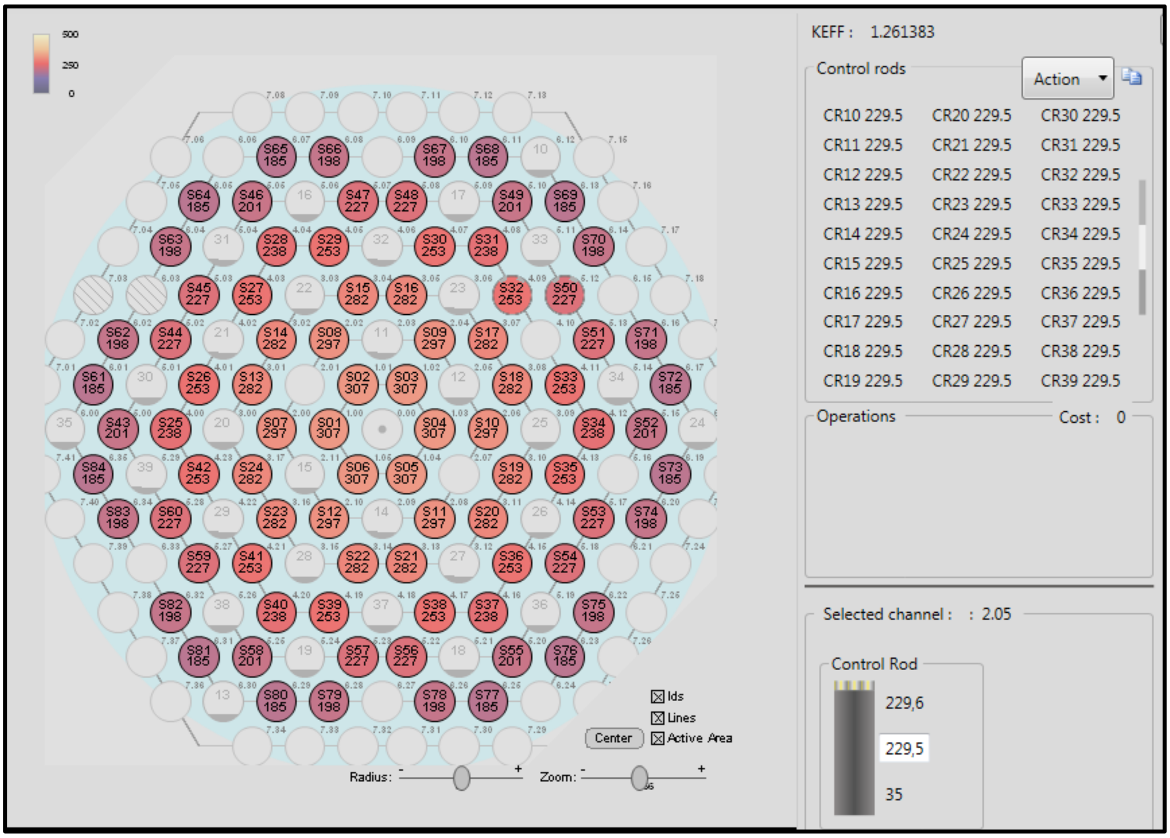


Figure 20. Example of core-map visualization (showing idealized HBWR core loading).

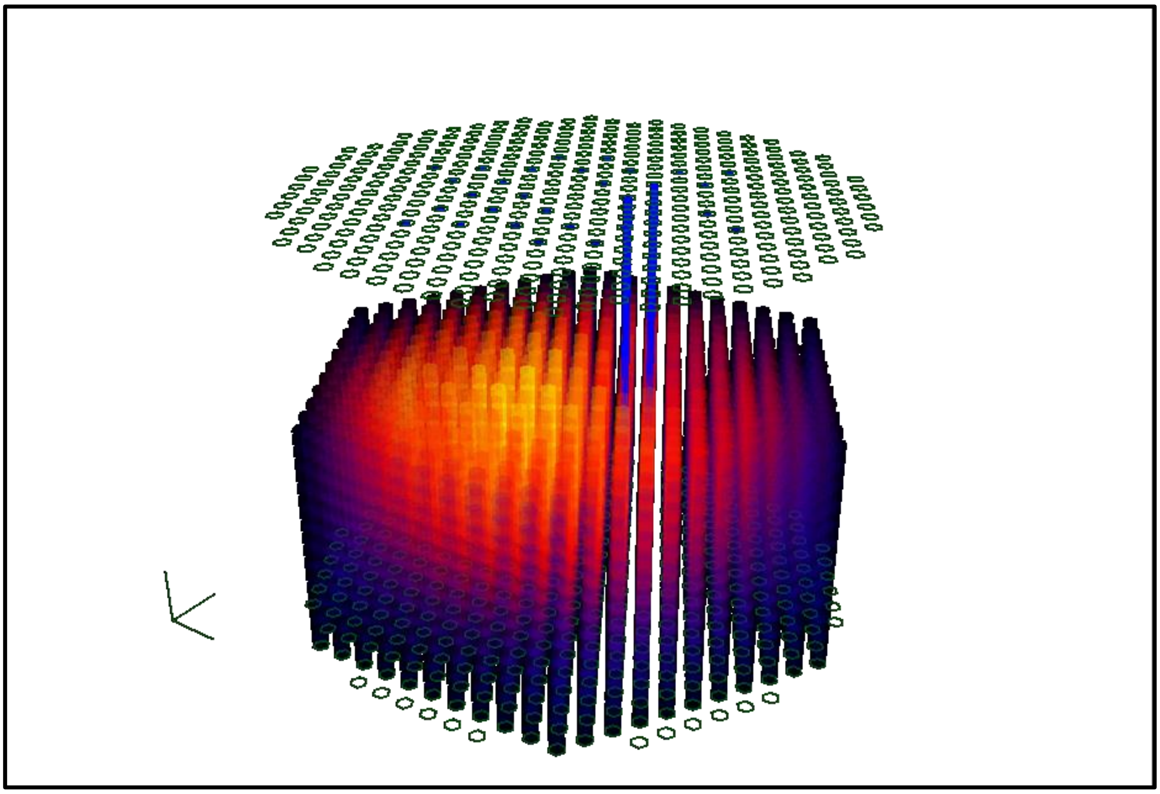


Figure 21. Example of nodal neutron flux visualization (Slice of HBWR idealized core with power tilt).

8 Conclusions

The intention in this phase of the project was to investigate different data solutions for nuclear reactor core simulators, specifically with the implementation of the HYBRID method.

In order to evaluate different solutions a set of requirements and metrics was required. This report starts the discussion as to what those requirements should be and attempts to define suitable cases for analysing performance.

There remains a lot of work in assessing individual database solutions against the requirements and more details are required for defining cases for analysis. However we have been able to determine the framework in which this work can proceed.

8.1 Further work

The data examples given here will be expanded to allow better assessment of their compliance with the stated requirements and use cases. These examples will cover an example from a typical BWR as well as the data specific models used to test the HYBRID method. The examples will cover different table based and document based solutions.

The data description in this report will also form the basis for developing criteria for selecting data for visualization and principals on how to visualize the data.

9 Acknowledgements

NKS conveys its gratitude to all organizations and persons who by means of financial support or contributions in kind have made the work presented in this report possible.

This work has benefited from concepts developed at IFE for the HBWR core management system. Many thanks to the efforts of Hans Olav Randem and Terje Lindén who have been instrumental in constructing that system.

Much of the early project work has been conducted by Tai Tien Huynh as part of his Master’s thesis at the Østfold University Collage. There we would also like to extend a thank you to his supervisor Øystein Haugen for useful discussions.

10 References

Fennern L. E., 2007, April, “ABWR Seminar – Reactor, Core & Neutronics”, GE Energy Nuclear.

Michel-Sendis F., Gauld I., Bossant M. & Soppera N.: 2014, October, *"A New OECD/NEA Database of nuclide compositions of spent nuclear fuel"*, Proceedings of the PHYSOR 2014 International Conference, Kyoto, Japan.
http://www.oecd-nea.org/sfcompo/

Nurseitov, N., Paulson M., Reynolds R. & Izurieta C., 2009 "Comparison of JSON and XML Data Interchange Formats: A Case Study." Caine 2009: 157-162.

Bibliographic Data Sheet **NKS-388**

Title	Data and visualization solutions for HYBRID core simulation method
Author(s)	William Beere[1], Tai Tien Huynh[2]
Affiliation(s)	[1]Institute for Energy Technology (IFE), Halden, Norway [2]Østfold University College (HiØ), Halden, Norway
ISBN	978-87-7893-474-1
Date	April 2017
Project	NKS-R / HYBRID
No. of pages	32
No. of tables	-
No. of illustrations	21
No. of references	3
Abstract max. 2000 characters	The modelling of neutron transport typically relies on two rather opposite approaches: the probabilistic approach, and the deterministic approach. The purpose of the present project is to combine both approaches in order to obtain fast running methods (thanks to the deterministic route) and accurate results (thanks to the probabilistic route). This so-called hybrid method will result in larger amounts of high-fidelity data than previous solutions to this problem. Viewing, comparing and storing this data should utilize the latest in data handling technology, covering input generation, data storage and output visualization. This report summarizes work performed so far in analysing the data aspects of this problem. This data system will not only be required to interface correctly with the proposed HYBRID method but will also have to interact with the envisaged user organization. At this stage of the project, the organizations are research institutes and universities. In the future, they may be reactor operators, fuel vendors or even reactor construction companies. Even further in the future spent fuel disposal companies may require some parts of the data solution. Considering these users we have proposed a list of requirements related to quality assurance, continuous development and aging management. This report makes a start at describing the data problem. Data types, uses and possible database configurations are discussed. Finally, some examples of different data structures are given and possible consequences investigated. The next project phase will focus on constructing and testing different data solutions and showing possible visualizations.
Key words	Neutron Transport, Database, SQL, NoSQL, Big Data

Available on request from the NKS Secretariat, P.O.Box 49, DK-4000 Roskilde, Denmark.
Phone (+45) 4677 4041, e-mail nks@nks.org, www.nks.org