Few group cross section representation based on Sparse Grid Methods

Danniëll Botes

Dissertation submitted in partial fulfilment of the requirements for the degree Master of Science (MSc) in Nuclear Engineering at the Potchefstroom campus of the North-West University

Supervisor: Dr P. Bokov
Co-supervisor: Prof E. Mulder

November 2012

Contents

List of Figures

List of Tables

Abstract

This thesis addresses the problem of representing few group, homogenised neutron cross sections as a function of state parameters (e.g. burn-up, fuel and moderator temperature, etc.) that describe the conditions in the reactor. The problem is multi-dimensional and the cross section samples, required for building the representation, are the result of expensive transport calculations. At the same time, practical applications require high accuracy. The representation method must therefore be efficient in terms of the number of samples needed for constructing the representation, storage requirements and cross section reconstruction time. Sparse grid methods are proposed for constructing such an efficient representation.

Approximation through quasi-regression as well as polynomial interpolation, both based on sparse grids, were investigated. These methods have built-in error estimation capabilities and methods for optimising the representation, and scale well with the number of state parameters. An anisotropic sparse grid integrator based on Clenshaw-Curtis quadrature was implemented, verified and coupled to a pre-existing cross section representation system. Some ways to improve the integrator's performance were also explored.

The sparse grid methods were used to construct cross section representations for various Light Water Reactor fuel assemblies. These reactors have different operating conditions, enrichments and state parameters and therefore pose different challenges to a representation method. Additionally, an example where the cross sections have a different group structure, and were calculated using a different transport code, was used to test the representation method. The built-in error measures were tested on independent, uniformly distributed, quasi-random sample points.

In all the cases studied, interpolation proved to be more accurate than approximation for the same number of samples. The primary source of error was found to be the Xenon transient at the beginning of an element's life (BOL). To address this, the domain was split along the burn-up dimension into "start-up" and "operating" representations. As an alternative, the Xenon concentration was set to its equilibrium value for the whole burn-up range. The representations were also improved by applying anisotropic sampling. It was concluded that interpolation on a sparse grid shows promise as a method for building a cross section representation of sufficient accuracy to be used for practical reactor calculations with a reasonable number of samples.

Keywords: sparse grids, Smolyak construction, few group neutron cross sections, parametrisation, cross section representation, hierarchical interpolation

Acknowledgements

I would like to express my gratitude towards my study leaders, Prof. Mulder and Dr. Bokov, for the support and assistance they offered. Others from the Radiation and Reactor Theory group at Necsa who I would like to thank are Dr. Wessel Joubert, for freely sharing his knowledge and experience, as well as for adapting the transport code whenever asked, and Mr. Rian H. Prinsloo, for assistance with programming and application design. Mrs. Hantie Labuschagne contributed her valuable experience in proofreading parts of this document and helping with the technical editing. Others that participated in proofreading are Ms. Nina Botes and Mr. Reinhardt Fourie from Universiteit Antwerpen. In addition, I would like to thank Dr. Vyacheslav G. Zimin from the National Research Nuclear University "MEPhI" in Moscow for performing the VVER pin cell calculations. The support of the National Research Foundation (NRF) to this project is also acknowledged.

Symbols and Abbreviations

Symbols

δ_{ij} Kronecker delta symbol

$\delta(x)$ Dirac delta function

$\zeta_j(x)$ test function

$\theta_i(x)$ Lagrange basis polynomial

$\Phi_g(\mathbf{r})$ heterogeneous flux in energy group g at position $\mathbf{r}$

$\phi_k(x)$ one-dimensional basis function from the space $C(\Omega)$

$\Sigma_{s,g}$ macroscopic cross section for reaction s in energy group g

$\sigma^i_{s,g}$ microscopic cross section for reaction s and nuclide i in energy group g

Ω closed interval that is a subset of $\mathbb{R}$, i.e. $\Omega = [a, b] \in \mathbb{R}$

$\mathcal{B}_0$ a subset of $\mathcal{P}_\infty$

$\mathcal{B}_1$ a subset of $\mathcal{P}_\infty$

$\mathcal{B}_\infty$ a subset of $\mathcal{P}_\infty$

$C(\Omega)$ vector space of all continuous functions on the domain Ω

$\mathcal{D}_l$ delta set, i.e. the set of discretisation nodes that are new to the grid of level l

d number of dimensions

E mathematical expectation

$\mathcal{H}_l$ the set of discretisation nodes that are in the grid of level l

$\mathbb{H}^d_q$ the set of discretisation nodes that are in the sparse grid of level q

$I[g(x)]$ integral operator acting on function $g(x)$

$L_n(x)$ Lagrange interpolation polynomial with n terms

l one-dimensional grid level

$\mathbb{N}$ set of natural numbers: $1, 2, 3, \ldots$

$N_i(\mathbf{r})$ number density of nuclide i at position $\mathbf{r}$

$\mathcal{O}$ Landau symbol – the rate of growth of a function

$P_i(x)$ Legendre polynomial of degree i

$\mathcal{P}_\infty$ enumerably infinite set of index vectors

$\mathcal{P}$ some finite subset of $\mathcal{P}_\infty$

$\mathcal{P}_{\text{TP}}$ the finite subset of $\mathcal{P}_\infty$ that is used for tensor product quadrature

$Q[g(x)]$ quadrature operator acting on function $g(x)$

$\Delta Q_l[g(x)]$ delta quadrature operator at hierarchical level l, acting on function $g(x)$

$\mathbb{Q}_q^d[g(\mathbf{x})]$ sparse grid quadrature operator acting on $g(\mathbf{x})$

$\mathbb{R}$ set of real numbers

$\mathbb{R}_+$ set of real numbers greater than 0

$r(x, \mathbf{c})$ residual function

s_i hierarchical surplus

$\mathcal{S}_\alpha$ set of admissible levels that is used to construct an anisotropic sparse grid

$\mathcal{S}_\Delta$ set of admissible levels that is used to construct a sparse grid by the Smolyak construction

$\mathcal{S}_{\text{com}}$ set of admissible levels that is used to construct a sparse grid by the combinatorial construction

$\mathcal{S}_\ell$ set of maximum one-dimensional levels

$\mathcal{S}_{\text{nest}}$ set of admissible levels that define the nodes that are new to a nested sparse grid of a given level

$U[g(x)]$ interpolation operator acting on function $g(x)$

$\Delta U_l[g(x)]$ delta interpolation operator at hierarchical level l, acting on function $g(x)$

$\mathbb{U}_q^d[g(\mathbf{x})]$ sparse grid interpolation operator acting on function $g(\mathbf{x})$

V volume of the node

Var variance

$W[g(x)]$ representation operator acting on function $g(x)$

w_i quadrature weight

$\mathbb{Z}$ set of integer numbers: $\ldots, -2, -1, 0, 1, 2, \ldots$

$\tilde{\mathbb{Z}}$ set of integer numbers greater than -2: $\{-1, 0, 1, 2, 3, \ldots\}$

$\otimes$ tensor product symbol

$\langle \cdot, \cdot \rangle$ inner product of two elements in a Hilbert space

Abbreviations

BOL Beginning Of Life, the state of an assembly before irradiation

ANOVA Analysis Of VAriance

HDMR High Dimensional Model Representation

HEADE HEterogeneous Assembly DEpletion code

HEU Highly Enriched Uranium

MCNP Monte-Carlo N-Particle transport code

MOX Mixed OXide fuel

MTR Materials Testing Reactor

PWR Pressurised Water Reactor

UNK Method of characteristics transport code

VVER Vodo-Vodyanoi Energetichesky Reactor, translated to English as Water-Water Power Reactor

Chapter 1

Introduction

1.1 Overview

1.1.1 Background and motivation

Homogenised few group neutron cross sections are widely used in reactor calculations. They depend on several material and thermo-hydraulic conditions in the reactor core and need to be represented as functions of these parameters, so that the reactor core simulator, where they are used, can reconstruct them at any core condition.

The cross section representation method currently employed at the South African Nuclear Energy Corporation (Necsa) requires some improvements. This method uses quadratic polynomials to approximate the cross section dependencies and treats burn-up as the base dependence. At each burn-up point, off-bases are calculated for each of the other state parameters, one parameter at a time. Error control is performed by sub-dividing the burnup range and fitting piece-wise quadratic polynomials over the sub-divided ranges until a pre-determined accuracy is achieved. The samples used to determine the accuracy are the same samples that are used for the fit, which limits the effectiveness of the error control mechanism.

The shortcomings of this approach include the inability to capture higher order off-base dependencies, a lack of information on interactions between two or more state parameters (cross terms) and the absence of a reliable error estimate. The inability to control the approximation error in a reliable and independent manner is a major deficiency. It was decided that a new cross section representation methodology that addresses these shortcomings should be developed. The new methodology should preferably also be flexible enough to handle possible future improvements in the calculational process, such as using extra state parameters. It should also provide some information on the relevance of the chosen parameters and the complexity of the dependence on any one dimension.

1.1.2 Purpose of the research

Sparse grid methods have been successfully applied to many problems in science and mathematics (Barthelmann, Novak, and Ritter, 2000; Everaars and Koren, 1997; Gerstner, 2007) and possess properties that could make them suitable to use for cross section representation, as will be discussed in later sections. The work presented in this thesis explores the possibilities that sparse grid methods offer, in terms of its application to cross section representation in a systematic manner. The utility of the method is demonstrated on several realistic examples and the possible improvements and adaptations are studied and compared.

It will be shown that sparse grid methods offer the ability to construct a representation that addresses the deficiencies of the current method used at Necsa, while sharing several positive aspects with other methods, such as:

- an accurate representation, with effective mechanisms for estimating and controlling the representation accuracy;
- information on the relevance of existing state parameters;
- information on the complexity of the dependence on any given state parameter; and
- the option of optimising the representation in terms of storage requirements and reconstruction time.

Some additional advantages of methods based on sparse grids are:

- an algorithm that is flexible in terms of the number of state parameters that it receives, without the need to change the code significantly for different reactor models;
- an algorithm that is scalable with the number of dimensions and can handle a large number of state parameters while remaining practical; and
- a highly efficient sampling strategy that may reduce the number of samples that are required to obtain a certain accuracy.

At the end of the study, a recommendation will be made regarding the way forward in terms of including a sparse grid based representation in Necsa's standard calculational path.

1.1.3 Outline of the dissertation

The rest of this introduction will explain some background about nuclear reactor calculations, followed by a discussion on few group neutron cross sections. At the end of the chapter the case for using sparse grid methods for cross section representation will be made.

In Chapter 2, the theoretical basis of the representation of functions will be laid out, first in one dimension and then in multiple dimensions. This will be followed by an overview of the theory relating to function representation in multiple dimensions using sparse grid methods.

Chapter 3 will contain details of the implementation of few group cross section representation on sparse grids, as well as some verification of the implementation.

Chapter 4 will give a description of the different reactor designs for which few group neutron cross section libraries were generated. This description will be followed by results on the accuracy of these libraries and demonstrate the strong and weak points of the method.

Chapter 5 will contain an analysis of the results shown in Chapter 4 and a discussion of the conclusions that can be drawn from them.

1.2 Reactor Simulation Methods

In the day to day running of a nuclear reactor, there are many instances where calculations, simulations or predictions could add value to the process. In some instances these simulations are critical to the safe operation of a nuclear reactor (Boyack et al., 1990). In other instances calculations help to improve the economic efficiency with which the reactor is managed. This may include optimisation studies to ensure efficient utilisation of resources, such as fuel elements, and maximising up-time. There are simulations that contribute to both the safety and the economical operation of the reactor, such as reload and core follow analyses (Müller et al., 1994). Reload calculations involve planning the core layout for the next operational cycle and ensuring that the core design chosen does not exceed certain safety parameters. Core follow calculations are used to keep track of the number densities of important isotopes during the operational cycle and also over the lifetime of any given element that contains fissile or fissionable material.

None of these calculations are simple or straightforward. Nuclear reactors are complex systems that require the modelling of the flow rates and temperature of fluids in the system, such as coolant (thermo-hydraulic analysis), as well as the neutron distribution in the core (neutronic analysis). There also exists feedback between thermo-hydraulic conditions and neutronic behaviour, thus coupled simulations are often required, which makes the problem very large and unwieldy. For the sake of focusing on what is relevant to this project, it is assumed that information about the thermo-hydraulic state of the reactor is available, e.g. a converged thermo-hydraulic solution exists.

The problem that remains to be examined is the time dependent neutron distribution in the core and surrounding structures. In order to find this neutron distribution, it is necessary to solve the Boltzmann transport equation — the equation which describes the distribution of the neutron population in space, energy and direction, and its evolution with time. The Boltzmann transport equation depends on quantities that are known as *cross sections*. These cross sections depend on the

material distribution in the reactor, which in its turn changes over time due to material interaction with the neutrons, a process that is commonly known as depletion. The equations that govern depletion are known as the Bateman equations. Following the evolution of the neutron distribution in time therefore requires solving the Boltzmann equation and the Bateman equations (Duderstadt and Hamilton, 1976).

A brief overview of the methods used to solve the Boltzmann transport equation follows.

1.2.1 Stochastic approach

The codes that use this approach are based on Monte Carlo methods, which were initially developed by Metropolis and Ulam (1949). Various transport codes based on the Monte Carlo method exist, such as MCNP (Monte Carlo N-Particle transport code) (Brown et al., 2002), Serpent (Leppänen, 2007) and TRIPOLI (Diop et al., 2007). MCNP is widely used, well supported and includes models of many physical phenomena that occur in reactors (Chibani and Li, 2002).

Monte Carlo methods are exact, under the conditions that the geometry is correctly treated and the cross sections that describe the particles' interaction with the material is accurate (Hébert, 2009). Additionally they are intuitive to understand and simple to implement (Spanier, 2010). A disadvantage is that they suffer from a slow convergence rate. This slow convergence rate makes the method computationally expensive (Geyer, 1994), thus obtaining an accurate static solution requires long run times, even on modern computers. Adding time dependence is possible and has been done for benchmarking other codes, but is not always feasible for providing timely support to ongoing reactor operations (Chang, 2005).

1.2.2 Deterministic approach

Several deterministic methods have been applied to solving the Boltzmann neutron transport equation, such as Collision Probabilities, Discrete Ordinates, Interface Current and Spherical Harmonics methods (Stacey, 2001). Most of these methods require some sort of discretisation of the problem domain, not only in space but also in energy. The energy domain is thus broken into several energy groups and cross sections are averaged over the interval that defines each group (Duderstadt and Hamilton, 1976).

The deterministic calculation path is often divided into a few steps, each separately tackling one area of complexity:

- The calculational path starts with what are known as evaluated data libraries. These are the result of extensive experimental and theoretical work and consist of values for cross sections of different interactions between particles. They include cross sections for interactions between

neutrons and nuclei at many different incident neutron energies, for most known nuclides (Chadwick et al., 2006).

- The nearly continuous data, available in the evaluated nuclear data libraries, are collapsed to a form that is referred to as fine energy groups (Leszczynski, Aldama, and Trkov, 2007).
- In the next step a so-called *lattice code*, also known as an *assembly code*, is used to collapse neutron cross sections from the fine group library to a *few group neutron cross section library*, which consists of between 1 and 18 energy groups, depending on the intended application (Trkov, 2000). Lattice codes solve the neutron transport equation for a smaller piece of the reactor core, usually one reactor element, for example, a fuel assembly. The impact of the rest of the core on the element is approximated by applying appropriate boundary conditions. The results are used for energy collapsing and, often, spatial homogenisation of neutron cross sections.
- The few group neutron cross sections are then passed to the *full core simulator*, where the reactor core is modelled in all three spatial dimensions, usually with thermo-hydraulic feedback (Knott and Yamamoto, 2010). Full core simulators routinely include time dependence.

In the final step above, the problem is often recast as a diffusion problem by making certain assumptions and simplifications and by applying Fick's law in order to do the full-core calculation. The derivation falls outside the scope of this text, but can be found in (Duderstadt and Hamilton, 1976). Making the diffusion approximation limits the applicability of the solution, but it makes calculations fast enough for many practical applications. Otherwise there are various simplifications that can be made to the transport equation that does not involve Fick's law (Stacey, 2001).

1.3 Homogenised Few Group Neutron Cross Sections

1.3.1 Place in the calculational path

As mentioned above, few group neutron cross sections are prepared as input to the full core simulator by lattice codes, which use neutron transport theory to group collapse and spatially homogenise fine group cross sections. Few group neutron cross sections depend on various parameters, known as *state parameters,* that describe the thermo-hydraulic and material properties of the reactor. Therefore, the ideal way to calculate them would be by embedding the lattice solution in the full core solution, so as to use the exact conditions in the core. This unfortunately adds the cost of repeated transport calculations every time a full core simulation is performed (Kosaka and Saji, 2000).

Alternatively, few group cross sections can be computed in advance, then stored in a library and reconstructed when necessary in the full core simulator. To retrieve the appropriate cross sections at any conditions in the core, the cross sections are represented in the library as a function of a given

set of state parameters. The choice of state parameters to use for this representation depends on the specific application, e.g. the particular type of reactor and the type of assembly in that reactor for which the model is produced. As an example, in Pressurised Water Reactors (PWRs), cross sections are often parametrised against burnup, fuel temperature, moderator temperature, moderator density and soluble boron concentration (Müller, Mayhue, and Zhang, 2007; Zimin and Semenov, 2005; Knott and Yamamoto, 2010). The number of state parameters are usually around five, and for different reactor designs may include thermal buckling, spectral index (the ratio between fast and thermal flux), void coefficient, or the number densities of important isotopes.

Some state parameters are instantaneous in nature, they only relate to the state in which the reactor is at a given moment in time, such as fuel and moderator temperatures or soluble Boron concentration. There are also a few state parameters that can be viewed as *history* parameters, these may include burnup and void history. History parameters are introduced in an attempt to capture the impact of previous changes in the reactor on the current reactor state (Dufek, 2011b; Knott and Yamamoto, 2010).

Two types of cross sections may be used in calculations. Microscopic cross sections represent the probability that two particles will interact in a specific manner, for example the probability that a given nucleus will absorb an incident neutron and then fission. A macroscopic cross section is the microscopic cross section of an isotope multiplied by its number density in a material (e.g. ^{235}U in UO_2 fuel) and describes the probability that a neutron undergoes a specific interaction in a given volume of material (Duderstadt and Hamilton, 1976).

An assembly code is used to calculate homogenised few group cross sections, that will be passed on to the full core simulator. It starts by calculating the heterogeneous flux in an assembly. This heterogeneous flux is then used to produce flux-volume weighted (homogenised) few group cross sections, as will be explained below.

In heterogeneous media, the macroscopic cross section is a function of position, since the number density of a nuclide is not constant throughout the volume. Homogenised macroscopic cross sections are a single effective cross section per energy group in a homogenised region for a given type of interaction. To calculate the homogeneous macroscopic cross section of reaction s in energy group g, where s may be absorption, fission, nu-fission or transport, the fine group macroscopic cross sections are flux weighted in the following way:

$$\bar{\Sigma}_{s,g} = \frac{\sum\limits_{j \in g} \int\limits_V \Sigma_{s,j}(\mathbf{r}) \Phi_j(\mathbf{r}) d\mathbf{r}}{\sum\limits_{j \in g} \int\limits_V \Phi_j(\mathbf{r}) d\mathbf{r}}, \tag{1.1}$$

where j is the j^{th} energy group in the fine group library and $\Phi_g(\mathbf{r})$ is the heterogeneous flux in energy group g at position $\mathbf{r}$. In some cases, the homogeneous macroscopic cross sections are the

only cross sections that are passed to the full core simulator.

There are some disadvantages to using only homogenised macroscopic cross sections. Firstly, the dependence of macroscopic cross sections on any given state parameter is more complex than the dependence of microscopic cross sections, especially the dependence on burnup (Müller, 1987). Constructing the library is therefore more difficult and it may contain significant errors. More importantly, though, it freezes the ratio of isotopes and therefore does not allow the full core simulator to account fully for changes to the spectrum due to core conditions that deviate from those used to produce the cross sections.

It is therefore preferable to use the transport code to also produce homogeneous microscopic cross sections for selected important isotopes, so-called *microscopic* materials. These are usually nuclides that are individually neutronically important, such as fissile and fissionable isotopes, fission products that have high capture cross sections as well as isotopes that are typically added as burnable absorbers. These microscopic cross sections are then also passed to the full core solver. Note that the microscopic materials still form only a subset of the materials that were used to perform the transport calculation. All other materials, such as structural material, are lumped into an effective macroscopic cross section.

The fine group library that is used in the transport code contains microscopic cross sections for many nuclides, typically around 200. The microscopic cross section for reaction s and nuclide i in energy group g is represented as $\sigma^i_{s,g}$. Homogenisation should preserve certain integral quantities, such as nuclear reaction rates. In the calculational system used for most of the problems presented in the Results, Chapter 4, the following homogenisation, which preserves reaction rates and assembly averaged number densities, was applied:

$$\bar{\sigma}^i_{s,g} = V \sum_{j \in g} \frac{\int_V \sigma^i_{s,j}(\mathbf{r}) N_i(\mathbf{r}) \Phi_j(\mathbf{r}) d\mathbf{r}}{\int_V N_i(\mathbf{r}) d\mathbf{r} \int_V \Phi_j(\mathbf{r}) d\mathbf{r}}, \tag{1.2}$$

where the integrals are taken over the volume per unit height of the assembly, j is the j^{th} group in the fine group library, $N_i(\mathbf{r})$ is the number density of isotope i at position $\mathbf{r}$ in the assembly and $\Phi_g(\mathbf{r})$ is the heterogeneous flux in energy group g at position $\mathbf{r}$.

1.3.2 Characteristics of few group cross sections

The specific problem of representing few group cross section dependencies on state parameters has some particularities that differ from the general representation problem. Some of these particularities make cross sections easier to represent, and some make them more difficult. This section will discuss the most pertinent points that impact cross section representation.

There are usually several state parameters that may influence a few group neutron cross section, both instantaneous and history parameters (Dufek, 2011b; Knott and Yamamoto, 2010; Zimin and Semenov, 2005). The representation of few group neutron cross sections is therefore a multi-dimensional problem.

The cross section dependencies may, however, not be equally complex in all the dimensions of the parameter space. Certain cross sections may exhibit a stronger dependence on some state parameters than on others, which may be characterised by the polynomial order needed to represent the dependence in that parameter in a polynomial representation, or a finer discretisation grid for a table representation. Including any prior knowledge that exists on which parameters are more prominent than others in a representation, such as by calculating more samples in the prominent dimension, may improve the accuracy. The ability to identify these complex dependencies as a result of the method would also be of value.

The set of cross sections at a given point in the parameter space, such as the cross sections for fission, absorption, scattering etc., are often the result of a single transport calculation, which allows them to be parametrised as a vector. This, however, limits the use of adaptive sampling techniques, since different cross sections may not share a primary dependence on the same parameter, or may have localised behaviour in different regions of the parameter space.

The transport calculation in itself is a complex numerical calculation that is performed with limited precision. This precision limits the accuracy that can be expected from the representation, and an insufficiently converged transport solution may introduce a large amount of uncertainty into the representation. Transport calculations are also computationally expensive, which limits the number of samples that are available for constructing the representation.

Another important consideration when constructing a cross section representation is that few group neutron cross sections are smooth functions of the state parameters, but they may have very localised features (Prinsloo et al., 2009; Zimin and Semenov, 2005), such as the Xenon transient at the beginning of an assemblies' life (BOL). Representations that use functions that cover the whole parameter space may miss such small-scale behaviour.

One of the state parameters that is often used as a history parameter is burnup, which is the change in material composition of an element over time as a result of irradiation. As mentioned before, burnup is governed by the Bateman equations, which are a stiff set of differential equations. Sampling of burnup is therefore not completely free, as under the constant flux or constant power assumptions, time steps are limited by the variation in the flux over the step (Duderstadt and Hamilton, 1976). Moreover, burnup can not be varied arbitrarily, as most instantaneous parameters can, and a calculation must be organised according to increasing burnup.

1.3.3 Current methods of representing few group neutron cross sections

Full core simulators may require data at any point in the parameter space, possibly at conditions other than those at which cross sections were calculated. This leads to a reconstruction problem: how to represent cross sections in a pre-computed library so that values can later be reconstructed efficiently and accurately at points between those that were initially calculated.

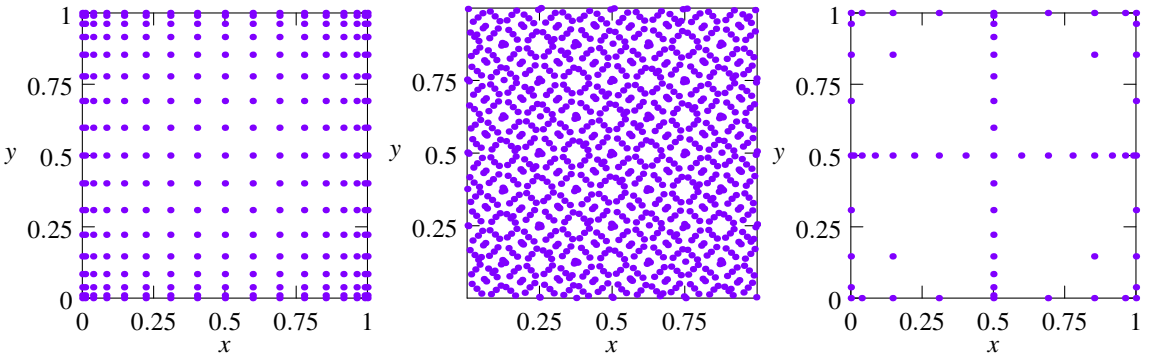
Assembly codes yield point-wise data – a single cross section value at a specific combination of the input parameters. This data is computationally expensive to obtain, therefore a practical limit is placed on the available number of samples. On the other hand, a high level of accuracy is required of the representation, in direct contrast to the computational constraints. The samples should be distributed in such a manner that the resulting representation covers all the foreseeable operational states of the reactor. This leaves us with two problems: how to optimally choose sample points in a multi-dimensional space and how to use these samples most efficiently.

These two problems may be addressed by considering three related aspects of representing any arbitrary function by another one. The first aspect is to determine which sampling scheme will be used. The second step is to choose a model on which to build the representation. When both of these tasks have been accomplished, the data of the samples must be converted into the information of the model, usually by determining the coefficients in some function expansion.

1.3.3.1 Sampling

There are many ways of sampling an arbitrary function in order to build a representation of that function. Some common schemes will be discussed in this section, and the sampling scheme that is proposed in this thesis will be explained.

A commonly used sampling scheme is the regular *tensor product mesh*, also known as a *Cartesian product grid*. It is constructed as a tensor product of regular one-dimensional meshes. Figure 1.1(a) is an example of a Chebyshev grid (see the definition in Section 2.1.3) that has been extended to two dimensions in this way. The number of nodes in any dimension is known as the order of the grid. The advantage of this scheme is that interpolation and approximation techniques that were developed for one dimension can be expanded to multiple dimensions in a straightforward way. It carries the disadvantage of the *curse of dimension*, a term first used by Bellman (1961) to describe the exponential growth in the number of samples required to maintain the same mesh spacing as the number of dimensions increases. As has already been discussed, a limited number of samples are available, but in addition to this, using the full tensor product grid may require large amounts of computer memory (Zimin and Semenov, 2005). This sampling scheme, using an equidistant one-dimensional grid, was used by Watson and Ivanov (2002) and Sato et al. (2010). It is the scheme that has traditionally been used for cross section representation by interpolation.



(a) Tensor Product Grid **(b) Low Discrepancy Grid** **(c) Sparse Grid**

Figure 1.1: Two-dimensional illustration of some sampling strategies

Another way to sample, is with uniformly distributed samples that are not based on a regular grid, e.g. Monte Carlo sampling and low discrepancy sequences. The latter are deterministic sequences that aim to provide the best possible coverage of the space with a finite number of samples. Low discrepancy grids are also known as quasi-random grids (Sobol', 1998) and an example can be seen in Figure 1.1(b). A number of authors have used these sequences, though using them for different purposes in the process of constructing a cross section representation: Zimin and Semenov (2005) and Dufek (2011b) for optimal coverage of the sample space and Bokov and Prinsloo (2007) as an efficient grid for computing multi-dimensional integrals when performing regression.

The sampling strategy that is used in this work is known as *sparse grid* sampling. Sparse grids were first described by Smoljak (1963) and are constructed as a union of low order tensor product grids. A two-dimensional example of such a grid can be seen in Figure 1.1(c). Sparse grid sampling has been extensively studied for the purpose of function representation in other fields (Reisinger and Wittum, 2007; Barthelmann, Novak, and Ritter, 2000; Griebel, 2006), but have only recently gained attention in cross section representation (Bokov, Botes, and Zimin, 2012). The reasons for choosing this sampling scheme are discussed in Section 1.4 and a fuller description of how they are constructed will follow in Section 2.3.

1.3.3.2 Model selection

The next step in the process of constructing a representation of few group homogenised neutron cross sections, is selecting the model, i.e. choosing the state parameters and an appropriate way to use the discrete samples that are available to construct a continuous representation. This includes deciding which functions to use for the representation and whether to use localised functions or ones that cover the whole parameter space.

One way of constructing the model is to choose state parameters at nominal conditions (default values for the state parameters), based on existing knowledge of the physical system, then to

perturb each state parameter from this nominal value. A Taylor series expansion based on these perturbations can then be used as a model of the cross section dependencies (Turski et al., 1997; Fujita et al., 2010). The expansion is usually of second order. It is known that low order Taylor series expansions are only valid in the close vicinity of the nominal conditions.

Another way, which is less problem dependent, is to use a linear combination of tensor products of one-dimensional basis functions. Depending on which basis functions are chosen, one could have either local or global support in the parameter space. If basis functions that are non-zero in only a sub-region of the parameter space are used, the support is local. If the chosen basis functions are non-zero throughout the space, the support is global. Watson and Ivanov (2002) and Botes and Bokov (2011) used tensor products of piece-wise linear basis functions. This basis function based approach is preferred when the goal is to construct a library suitable for transient calculations, since transient libraries require large intervals of variation of the state parameters.

Considerations from the field of high dimensional model representation may be used to decrease the effective dimensions of the problem. This approach will be discussed further in Section 2.4, and Eq. (2.95) is pertinent to this discussion. Any function that depends on multiple input parameters can be decomposed into a sequence of functions with fewer variables. Some functions may have low-order correlations between the input variables and the decomposition can then be truncated to a sequence of functions with fewer dimensions than the original problem (Rabitz et al., 1999).

1.3.3.3 Fitting the model against the samples

Once the model has been chosen, one needs to find a way to apply the model to the specific problem under consideration. This may be done by determining the coefficients in a Taylor series expansion or a model built on tensor products of one-dimensional basis functions, as discussed above. There are traditionally two ways of doing this, *approximation* and *interpolation*.

Approximation was done via step-wise regression by Zimin and Semenov (2005) and via quasi-regression by Bokov and Prinsloo (2007). Dufek (2011a) did the same via standard regression. All of these approaches can use arbitrary samples, therefore any of the sampling schemes discussed in Section 1.3.3.1 are suitable.

The simplest form of interpolation is a table representation of the samples with a rule for interpolating with linear functions between them, e.g. Watson and Ivanov (2002). The advantages of this method is that it is simple to implement and to reconstruct from. It also produces a library that is easily readable and interpretable for a human, since it contains the explicit cross section values at particular points in the state parameter space along with a rule for interpolation.

One can also use higher order polynomials or trigonometric functions, or a mixture of different basis functions in different directions (Guimaraes and Müller, 2009). In this work interpolation with higher order Lagrange polynomials and quasi-regression with higher order Legendre polynomials

were used. These methods will be discussed in more detail in Chapter 2.

1.3.4 Constructing a library

There are two separate, but related issues regarding the construction of a cross section library:

1. identifying relevant state parameters, which is currently done by hand and requires expert knowledge and experience; and
2. representing cross sections in terms of these parameters.

Identifying the relevant state parameters is a matter of trial and error and there is no standard procedure for this process. This work addresses the second aspect.

Some criteria by which a cross section representation may be judged are:

Library size. There may be many assemblies/material regions in a reactor and a library needs to be generated for each of these. The library needs to contain cross sections for several reaction types and cover multiple energy groups.

Library creation time. This consideration has two components:

> *Number of lattice code runs.* On the one hand, sampling is expensive and the number that can reasonably be computed is limited. On the other hand, a library only needs to be created once for any new element type during normal reactor operations. If, however, one intends to do design studies, many libraries may need to be constructed.
>
> *Approximation construction time.* The library construction process itself may be computationally expensive, even after all the necessary samples have been taken. This should be considered when evaluating the available options.

Controlling the accuracy of the representation. Some knowledge of, and strategy to limit the approximation error, is important for neutron cross section representation. Local error control would be preferable, but at least a global error control strategy is necessary.

Cross section reconstruction time. Reconstruction usually happens in the full-core simulator, where calculation speed is an important consideration.

1.4 Motivation for Using Sparse Grid Methods

As a result of the curse of dimension, discussed in Section 1.3.3.1, adding extra state parameters to the representation model would negatively impact the construction time if any standard tensor

product based approach is used. Sampling methods, e.g. Monte Carlo, on the other hand converge relatively slowly, even though the convergence is often not dependent on the number of dimensions (Mathé and Novak, 2007). For this reason they only provide an advantage for very high dimensional problems.

Sparse grid methods aim to optimise the convergence rate of the representation error for certain classes of functions, specifically smooth functions. Since sparse grids are a union of standard tensor product grids, they retain the advantage that, if an algorithm is known for solving the problem in one dimension, an algorithm for solving the problem for an arbitrary number of dimensions is completely determined by tensor products of the one-dimensional algorithm. In this study, the problem under consideration is the representation of a smooth function in a moderate number of dimensions (four to eight). The literature indicates that sparse grid methods should be well suited to a problem with these characteristics (Novak and Ritter, 1996; Burkardt, Gunzburger, and Webster, 2007; Nahm, 2005; Everaars and Koren, 1997; Griebel and Hamaekers, 2007; Wasilkowski and Wozniakowski, 1995).

The aim of this work is then to combine the efficiency and flexibility of sparse grid sampling with the advantages of existing representation models, such as approximation by regression or interpolation. It will also be shown that dimension-wise adaptivity make sparse grids well suited to a problem such as cross section representation that is more complex in some dimensions than in others. Given all of these potential advantages, applying them to cross section representation seems to be a worthy endeavour.

Chapter 2

Theoretical Background

Section 1.3.1 discussed the need to build a representation, either an approximation or an interpolation, of few group neutron cross section dependencies on the relevant state parameters. In this chapter, we will review the theory of function representation, first in one dimension, then in multiple dimensions. Finally, we will discuss the method that was used in this work as well as the model reduction techniques that were employed.

2.1 Function Representation in One Dimension

This section will give a brief overview of the theory of representation in one dimension that will be applicable to the methods used in this work. It will serve as a foundation for the discussion on function representation in multiple dimensions, which will follow after. The discussion in this section will closely follow that of Boyd (2000).

We assume that the function that we want to represent, $f(x)$, is continuous on the compact subspace $x \in \Omega = [a, b] \subset \mathbb{R}$, i.e. $f(x) \in C(\Omega)$ where $C(\Omega)$, is the space of all functions that are continuous over Ω. $C(\Omega)$ has a useful property: it is complete. In other words, any function in it can be decomposed as

$$f(x) = \sum_{i=1}^{\infty} c_i \phi_i(x), \tag{2.1}$$

where the $\phi_i(x)$ form a infinite, enumerable set of linearly independent basis functions that span $C(\Omega)$ (DeVore and Lorentz, 1993). We approximate the original function by truncating the infinite series to n terms in the following way:

$$f(x) \approx f_n(x; \mathbf{c}) = \sum_{i=1}^{n} c_i \phi_i(x), \tag{2.2}$$

where the finite, enumerable set $\{\phi_i,\, i = 1, 2, \ldots, n\}$ is a subset of $\{\phi_i,\, i = 1, 2, \ldots, \infty\}$ and $\mathbf{c} = (c_1, \ldots, c_n)$ are unknown coefficients. The completeness property mentioned above implies that one can obtain an approximation that is arbitrarily close to the original function by increasing n. For functions on a compact subspace of $\mathbb{R}$, polynomials are often chosen as the basis functions (DeVore and Lorentz, 1993). To 'maximise' the independence of the basis functions, orthogonal polynomials are often used.

There are several ways to determine the unknown coefficients and the properties that are required of the basis functions. Most of these come from the so-called 'method of mean weighted residuals', which give a framework that enables one to choose a distribution of the representation error over Ω. A short discussion of the pertinent elements of mean weighted residual methods follows below, Boyd (2000) gives a more complete overview for the interested reader.

2.1.1 Method of mean weighted residuals

Let us define the residual function as

$$r(x; \mathbf{c}) = f(x) - f_n(x; \mathbf{c}) = f(x) - \sum_{i=1}^{n} c_i \phi_i(x), \tag{2.3}$$

where $\mathbf{c} = (c_1, \ldots, c_n)$ are a set of coefficients that are as yet undetermined. Note that the residual is a function of the independent variable, x, as well as the coefficients, c_i. The objective is to find a representation that will minimise $r(x; \mathbf{c})$ in some sense.

We introduce the *inner product* of two functions, defined as

$$\langle u(x), v(x) \rangle = \int_{\Omega} u(x) v(x) dx, \tag{2.4}$$

where both $u(x)$ and $v(x)$ are elements of $C(\Omega)$ and $\Omega = [a, b] \subset \mathbb{R}$. If $u(x)$ and $v(x)$ are orthogonal then

$$\langle u(x), v(x) \rangle = \begin{cases} a_{uv}, & u(x) = v(x), \\ 0, & u(x) \neq v(x), \end{cases} \tag{2.5}$$

with a_{uv} a scaling factor. If $u(x)$ and $v(x)$ are orthonormal, then $a_{uv} = 1$.

The mean weighted residual method determines the coefficients by imposing the conditions

$$\langle \zeta_j(x), r(x; \mathbf{c}) \rangle = 0 \tag{2.6}$$

for a given set of *test functions* $\{\zeta_j(x), j = 1, 2, \ldots, n\}$. Conditions (2.6) lead to a system of n linear equations with the n coefficients, c_i, as the unknowns. This system of linear equations can be obtained by substituting the definition of the residual function, Eq. (2.3), into Eq. (2.6). By using

the properties and the definition of the inner product, see Eq. (2.4), the inner product of the test function and the remainder may be decomposed as

$$\begin{aligned}\langle \zeta_j(x), r(x;\mathbf{c})\rangle &= \langle \zeta_j(x), f(x) - f_n(x;\mathbf{c})\rangle \\ &= \langle \zeta_j(x), f(x)\rangle - \langle \zeta_j(x), f_n(x;\mathbf{c})\rangle \\ &= \langle \zeta_j(x), f(x)\rangle - \sum_{i=1}^{n} c_i \langle \zeta_j, \phi_i(x)\rangle.\end{aligned} \tag{2.7}$$

Applying the conditions from Eq. (2.6) to the above decomposition produces the linear system in the form

$$\int_{\Omega} \zeta_j(x) f(x) dx = \sum_{i=1}^{n} c_i \int_{\Omega} \zeta_j(x)\phi_i(x) dx, \tag{2.8}$$

where $j = 1, 2, \ldots, n$.

A variety of appropriate test functions exist, and the choice of the test function will determine the distribution of the error. We will consider two test functions, which will lead us to the methods of *collocation* and *least squares approximation*. A short explanation of how these methods are used will follow.

2.1.1.1 Collocation

Let us choose $\zeta_j(x) = \delta(x - x_j)$ as our set of test functions, where $\{x_j,\, j = 1, 2, \ldots, n\}$ are known as "interpolation" or "collocation" points and $\delta(x)$ is the Dirac delta function. From Eq. (2.8), it follows that

$$\int_{\Omega} \delta(x - x_j) f(x) dx = \sum_{i=1}^{n} c_i \int_{\Omega} \delta(x - x_j)\phi_i(x) dx, \tag{2.9}$$

which leads to

$$f(x_j) = \sum_{i=1}^{n} c_i \phi_i(x_j). \tag{2.10}$$

Solving this set of linear equations for any given set of basis functions will determine $\{c_i,\, i = 1, 2, \ldots, n\}$ and therefore $f_n(x;\mathbf{c})$. If we require that $\phi_i(x_j) = \delta_{ij}$, where δ_{ij} is the Kronecker delta symbol, then $\phi_i(x)$ is known as a *cardinal* function (Boyd, 2000). In this case, the linear equations reduce to $f(x_j) = c_j$. Function representation as formulated in Eq. (2.2) then becomes

$$f_n(x;\mathbf{c}) = \sum_{i=1}^{n} f(x_i)\phi_i(x) \tag{2.11}$$

when collocation, also known as interpolation, is used.

2.1.1.2 Least squares approximation

Let us use $\zeta_j(x) = \phi_j(x)$, $(j = 1, \ldots, n)$ as an alternative set of test functions. Substituting them into Eq. (2.8) yields

$$\int_\Omega f(x)\phi_j(x)dx = \sum_{i=1}^{n} c_i \int_\Omega \phi_i(x)\phi_j(x)dx. \tag{2.12}$$

It can be shown that this choice of $\zeta_i(x)$ is equivalent to minimising the square of the L_2 norm of the residual function, $||r(x)||_2^2 = \langle r(x), r(x)\rangle$, with respect to c_i. To find the coefficients that correspond to this minimum, we first expand the inner product of the residual function

$$\begin{aligned}\langle r(x), r(x)\rangle &= \int_\Omega [r(x)]^2\, dx = \int_\Omega [f(x) - f_n(x;\mathbf{c})]^2\, dx \\ &= \int_\Omega [f(x)]^2\, dx - 2\int_\Omega [f(x)f_n(x;\mathbf{c})]\, dx + \int_\Omega [f_n(x;\mathbf{c})]^2\, dx\end{aligned} \tag{2.13}$$

We then minimise Eq. (2.13) by taking the first partial derivative with respect to each coefficient c_i and setting it equal to zero,

$$\int_\Omega \frac{\partial}{\partial c_i}[f(x)]^2\, dx - 2\int_\Omega \frac{\partial}{\partial c_i}[f(x)f_n(x;\mathbf{c})]\, dx + \int_\Omega \frac{\partial}{\partial c_i}[f_n(x;\mathbf{c})]^2\, dx = 0. \tag{2.14}$$

Taking the derivative of $f_n(x, \mathbf{c})$ with respect to c_i, we obtain

$$\frac{\partial}{\partial c_i}\sum_{k=1}^{n} c_k\phi_k(x) = \sum_{k=1}^{n}\phi_k(x)\frac{\partial c_k}{\partial c_i} = \sum_{k=1}^{n}\phi_k(x)\delta_{ik} = \phi_i(x) \tag{2.15}$$

and substituting Eq. (2.15) into Eq. (2.14) leads to

$$\int_\Omega [\phi_i(x)f(x)]\, dx = \int_\Omega \left[\phi_i(x)\sum_{k=1}^{n} c_k\phi_k(x)\right] dx = \sum_{k=1}^{n} c_k \int_\Omega [\phi_i(x)\phi_k(x)]\, dx, \tag{2.16}$$

which corresponds to the system of linear equations in Eq. (2.12). If we require the basis functions to be orthogonal the following holds:

$$\langle \phi_i(x), \phi_k(x)\rangle = \delta_{ik}||\phi_i(x)||_2 \cdot ||\phi_k(x)||_2, \tag{2.17}$$

thus Eq. (2.16) becomes

$$\int_\Omega [\phi_i(x)f(x)]\, dx = \sum_{k=1}^{n} c_k\delta_{ik}||\phi_i(x)||_2 \cdot ||\phi_k(x)||_2 = c_i||\phi_i(x)||_2^2. \tag{2.18}$$

The formula for representation by least squares approximation is thus

$$f(x) = \sum_{i=1}^{n} \frac{\langle f(x), \phi_i(x) \rangle}{\langle \phi_i(x), \phi_i(x) \rangle} \phi_i(x). \tag{2.19}$$

The coefficients can be found by calculating the following integrals

$$c_i = \frac{\int_{\Omega} [\phi_i(x) f(x)] \, dx}{\int_{\Omega} [\phi_i(x)]^2 \, dx} = \frac{\langle f(x), \phi_i(x) \rangle}{\langle \phi_i(x), \phi_i(x) \rangle}. \tag{2.20}$$

Therefore, in order to construct a least squares approximation, the integrals in Eq. (2.20) have to be evaluated. Since $f(x)$ is assumed to be only known at some sample points, a numerical quadrature should be used to estimate the regression coefficients, c_i.

Orthonormal basis functions can be used to alleviate the computational burden by eliminating the calculation of the scaling function. If orthonormal basis functions are used, $\langle \phi_i(x), \phi_i(x) \rangle = 1$, and Eq. (2.19) can be simplified to

$$f(x) = \sum_{i=1}^{n} \langle f(x), \phi_i(x) \rangle \phi_i(x). \tag{2.21}$$

Using an approximation in the form of Eq. (2.21) is known as quasi-regression (An and Owen, 2001) and will be used in this work as the basis for constructing the multi-dimensional approximation of cross section dependencies. The problem of representing a function by approximation may therefore be made equivalent to the problem of numerically integrating the product of that function and the basis function, using only the value at some sample points. Numerical integration will be discussed further in Section 2.1.5.1.

2.1.2 Polynomial basis functions

In both of the above instances, it is necessary to choose an appropriate set of basis functions. There are several classes of polynomials that satisfy the requirements of different mean weighted residual methods. Amongst others, Legendre polynomials, Chebyshev polynomials and Hermite polynomials are all orthogonal, but on different intervals and with respect to different weights. Legendre polynomials satisfy the orthogonality condition, Eq. (2.17), and are widely used for least squares approximation. Lagrange basis polynomials (see below), by their construction, are cardinal functions and are therefore suitable basis functions for interpolation. In this work, we use Legendre polynomials as basis functions for approximation and Lagrange basis polynomials as basis functions for interpolation. All the basis functions were rescaled to $\Omega \to \mathcal{I} = [0, 1]$ in this work. Rescaling of the domain is discussed as and where necessary.

Legendre polynomials are solutions to Legendre's differential equation and can be constructed by the recursion formula (Churchhouse et al., 1981)

$$(i+1)P_{i+1}(x) = (2i+1)xP_i(x) - iP_{i-1}(x), \tag{2.22}$$

with the first two polynomials in the series defined as

$$P_0(x) = 1,\ P_1(x) = x.$$

Legendre polynomials are defined on the interval $[-1, 1]$, where they are orthogonal with weight 1. They satisfy the orthogonality condition:

$$\int_{-1}^{1} P_i(x)P_k(x)dx = \frac{2}{2i+1}\delta_{ik}. \tag{2.23}$$

After rescaling to the interval $[0, 1]$ and normalising, the basis functions that were used in our work for polynomial approximation are

$$\phi_i(x) = \sqrt{2i+1}P_i(2x-1). \tag{2.24}$$

The Lagrange interpolation polynomial is defined as (Churchhouse et al., 1981)

$$L_n(x) = \sum_{i=1}^{n} f(x_i)\theta_i(x) \tag{2.25}$$

where

$$\theta_i(x) = \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{2.26}$$

are polynomials of degree $n-1$ and have the property $\theta_i(x_j) = \delta_{ij}$, i.e. they are cardinal functions. The $\theta_i(x)$ are also known as the *Lagrange basis polynomials* (Boyd, 2000). The equation for interpolation using Lagrange basis functions is therefore simply $f_n(x) = L_n(x)$.

The one-dimensional basis functions can be extended to multiple dimensions

2.1.3 Selecting the sampling scheme

Now that the basis functions have been chosen, we can proceed to the task of determining the values of the representation coefficients. Both interpolation and quadrature require samples of the function being represented at certain, as yet undetermined, points, x_i $(i = 1, \ldots, n)$, also called *nodes* or *abscissae*.

In this work we used nodes that are based on the endpoints and extrema of the Chebyshev

polynomials of the first kind, known in various sources as Chebyshev nodes, Gauss-Lobatto nodes, or as the nodes of the Clenshaw-Curtis quadrature rule (Boyd, 2000). They were used as samples for both interpolation and quadrature and will henceforth be referred to as *Chebyshev nodes*, or the *Chebyshev grid*. Using the Chebyshev nodes, which are more closely packed at the edges of the interval than at the centre, is known to mitigate Runge's phenomenon, the tendency of high degree polynomial interpolants to oscillate with growing amplitude as the boundary is approached (Berrut and Trefethen, 2004).

Let us specify the number of nodes in such a grid as

$$n_l = \begin{cases} 2^{l-1} + 1, & l > 1, \\ 1, & l = 1, \end{cases} \tag{2.27}$$

where l is called the *level* of the grid and $l \in \mathbb{N}$. Other sources may use a different convention whereby l starts from 0 (Reisinger and Wittum, 2007; Bokov, Botes, and Zimin, 2012), but in this work we use the convention $l \geq 1$. Parameter l is known as the *accuracy level* of the quadrature or interpolation rule for which the grid is used. The number of nodes in the grid, n_l, is also known as the *order* of the quadrature or interpolation rule.

The Chebyshev nodes are defined on the interval $[-1, 1]$, but were rescaled in this work to the interval $[0, 1]$ for reasons that will become clear in later sections. We introduce the set of Chebyshev nodes, $\mathcal{H}_l$, and define it as all the nodes in a Chebyshev grid of level l. The abscissae that belong to $\mathcal{H}_l$ are, after rescaling to $[0, 1]$, defined as

$$x_{i,l} = \begin{cases} \frac{1}{2}\left[1 - \cos\left(\frac{i-1}{n_l - 1}\pi\right)\right], & 1 \leq i \leq n_l \text{ and } n_l > 1, \\ \frac{1}{2}, & n_l = 1 \Rightarrow i = 1. \end{cases} \tag{2.28}$$

Note that the node has two indices: index i labels the abscissa of the node from $i = 1$ for $x = 0$ to $i = n$ for $x = 1$, and index l indicates the level at which the node carries index i. The reason for the double indexing will become apparent in the next section. This definition is equivalent to an equidistant mesh, defined on a semicircle and projected onto a straight line. The semi-circle is divided into arcs subtended equal angles, by successively bisecting the angles. This projection from the semi-circle to the real number line is illustrated in Figure 2.1.

2.1.4 Hierarchical construction of the grid

Chebyshev nodes, as defined by Eqs. (2.27) and (2.28), are nested, that is for any l the following holds: $\mathcal{H}_l \subset \mathcal{H}_{l+1}$. This allows us to define the *delta set* $\mathcal{D}_l$ as the set of nodes that are new to level l. That is

$$\mathcal{D}_l = \mathcal{H}_l \setminus \mathcal{H}_{l-1}. \tag{2.29}$$

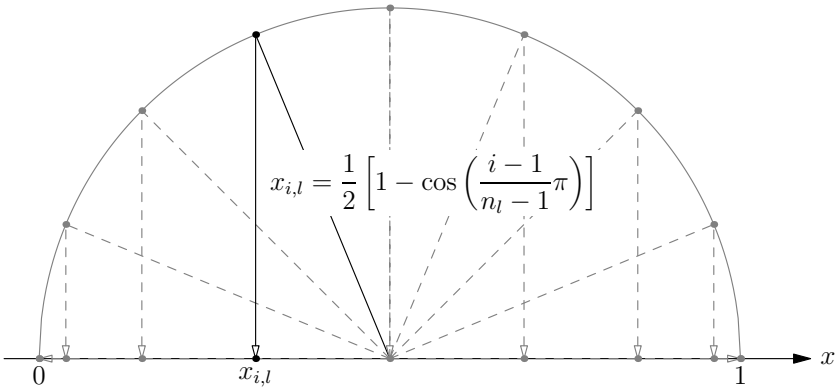


Figure 2.1: Grid obtained by projecting equally spaced points on the semi-circle onto the line of real numbers

We assign to any point in the grid its own level, if the point first appears at level $1 \leq j \leq l$, then that point has level j. There are thus two levels that can be discussed, the level of the grid, denoted by l and the level of a given point, denoted by j. If a point forms part of the delta set $\mathcal{D}_l$, then $j = l$.

Grid $\mathcal{H}_l$ can therefore be presented as the union of non-overlapping delta sets

$$\mathcal{H}_l = \bigcup_{1 \leq j \leq l} \mathcal{D}_j. \tag{2.30}$$

2.1.5 Determining the coefficients of the representation

2.1.5.1 Numerical integration on a hierarchically constructed grid

Consider the problem of numerically evaluating a definite integral of an arbitrary function $g(x)$,

$$I[g(x)] = \int_{\Omega} g(x)dx. \tag{2.31}$$

The integral operator, $I[g(x)]$, in Eq. (2.31) can be approximated with a quadrature:

$$I[g(x)] \approx Q[g(x)], \tag{2.32}$$

where $Q[g(x)]$ is the quadrature operator[1] that operates on $g(x)$. It is defined by

$$Q[g(x)] = \sum_{i=1}^{n} w_i g(x_i), \tag{2.33}$$

[1]Integral and quadrature operators act as linear functionals, and are referred to as operators in the sense that functionals are a special case of operators.

with the w_i known as quadrature weights, the x_i are some, as yet undetermined, points at which $g(x)$ is sampled and n is the order of the rule. For least squares approximation, the integrand is $g_i(x) = \phi_i(x)f(x)$, see Eq. (2.20).

Several quadrature methods are available, such as Newton-Cotes, Clenshaw-Curtis, Gauss-Lobatto, etc. The quadrature rule that is used dictates both the weights, w_i, and the coordinates, x_i, of the nodes at which $g(x)$ is sampled. As an example, the Newton-Cotes quadrature rule requires equidistant nodes.

The Clenshaw-Curtis method was employed as the quadrature rule in this work. It is known as an *interpolatory* quadrature and obtains the weights by interpolating $g(x)$ with the Chebyshev polynomials as basis functions and the Chebyshev nodes as samples. The resulting polynomial is then integrated analytically. The Clenshaw-Curtis quadrature with n samples can accurately integrate a polynomial of degree $n-1$. This quadrature requires the use of the Chebyshev nodes, defined in Section 2.1.3 and the weight that corresponds to each abscissa x_i is (Gerstner and Griebel, 1998)

$$w_{i,l} = \begin{cases} \dfrac{1}{2n_l\,(n_l-2)}, & i = 1 \text{ or } i = n_l, \\ \dfrac{1}{n_l-1}\left(1 + 2\,{\sum_{k=1}^{(n_l-1)/2}}'\,\dfrac{1}{1-4k^2}\cos\left(\dfrac{2\pi(i-1)k}{n_l-1}\right)\right), & 2 \le i \le n_l - 1, \end{cases} \tag{2.34}$$

where the prime on the summation sign indicates that the last term is halved. The weights in Eq. (2.34) correspond to the abscissa that have been scaled to $[0, 1]$.

Let us denote the quadrature operator that uses samples from a grid of level l, as $Q_l[g(x)]$. Eq. (2.33) is now written as

$$Q_l[g(x)] = \sum_{i=1}^{n_l} w_{i,l}\, g(x_{i,l})$$

and we define the zero-order approximation as $Q_0[g(x)] = 0$. This allows us to define a *delta quadrature operator* as

$$\Delta Q_l[g(x)] = Q_l[g(x)] - Q_{l-1}[g(x)] \tag{2.35}$$

for a given level l. The quadrature operator of level l may then be constructed as the sum of the delta quadrature operators of all levels less than or equal to l,

$$Q_l[g(x)] = \sum_{j=1}^{l} \Delta Q_j[g(x)]. \tag{2.36}$$

Eq. (2.36) presents the quadrature operator in so-called *hierarchical* form, which will be used in later descriptions of the specific implementation of quadrature on sparse grids.

Given a quadrature rule that is based on a nested grid, in order to calculate the quadrature weights

in the hierarchical representation, it is necessary to find the nodes that were inherited from previous levels. Where a node from level l, $x_{i,l}$ coincides with a node from level $l-1$, $x_{k,l-1}$ the difference in their weights gives the weight of the delta quadrature operator in that node (Gerstner and Griebel, 1998). The following explicit form of the delta quadrature operator can then be obtained

$$\begin{aligned}\Delta Q_l[g(x)] &= \sum_{i=1}^{n_l} w_{i,l} g(x_{i,l}) - \sum_{k=1}^{n_{l-1}} w_{k,l-1} g(x_{k,l-1}) \\ &= \sum_{i=0}^{2^{l-2}} w_{2i+2,l} g(x_{2i+2,l}) + (w_{2i+1,l} - w_{i+1,l-1}) g(x_{2i+1,l})\,.\end{aligned} \tag{2.37}$$

From the above one can see that even numbered nodes are new to the level and have their weight calculated only for that level, whereas odd-numbered nodes have appeared in the previous level, and this has to be accounted for when their weight is updated. This description will become more clear when the application of the hierarchical construction of the quadrature is discussed in Chapter 3.

2.1.5.2 Interpolation on a hierarchically constructed grid

Interpolation may also be recast in the hierarchical form, which will be of use later on in the discussion on sparse grid interpolation. This process will first be discussed for the general case of interpolation and then applied to Lagrange interpolation as discussed in Section 2.1.2.

Let us introduce the interpolation operator that acts on a given function $f(x)$, on a Chebyshev grid of level l, as

$$U_l[f(x)] = \sum_{i=1}^{n_l} f(x_{i,l}) \phi_{i,l}(x), \tag{2.38}$$

and state that $U_0[f(x)] = 0$. It is then possible to define the *delta interpolation operator*,

$$\Delta U_l[f(x)] = U_l[f(x)] - U_{l-1}[f(x)], \tag{2.39}$$

which is simply the difference between the interpolation at the current level and the interpolation at the previous level. Klimke (2006) has shown that the interpolation operator applied to the previous level interpolation is exact, that is

$$U_{l-1}[f(x)] = U_l[U_{l-1}[f(x)]]. \tag{2.40}$$

Substituting Eq. (2.40) into Eq. (2.39) allows us to write the delta interpolation operator as

$$\Delta U_l[f(x)] = U_l[f(x)] - U_l[U_{l-1}[f(x)]]. \tag{2.41}$$

Since $U[f(x)]$ is linear, it is possible to express the delta interpolation operator as

$$\Delta U_l[f(x)] = U_l\left[f(x) - U_{l-1}[f(x)]\right]. \tag{2.42}$$

By substituting the definition of the interpolation operator, Eq. (2.38), into Eq. (2.42), the delta interpolation operator can be written as

$$\Delta U_l[f(x)] = \sum_{i=1}^{n_l} \left(f(x_{i,l}) - U_{l-1}[f(x_{i,l})]\right) \phi_{i,l}(x). \tag{2.43}$$

This form closely resembles the form of the standard interpolation operator, Eq. (2.38), with the quantities

$$s_{i,l} = f(x_{i,l}) - U_{l-1}[f(x_{i,l})] \tag{2.44}$$

as coefficients, instead of the function value. The $s_{i,l}$ are known as *hierarchical surpluses* (Klimke, 2006) and quantify the difference between the interpolation at the previous level and the function value at nodes that are new to level l. Given that the interpolation is performed on the nested Chebyshev grid, $f(x_{i,l}) = U_{l-1}[f(x_{i,l})]$ for all $x_{i,l} \in \mathcal{H}_{l-1}$ i.e. for all nodes that have node level $j \leq l-1$. Therefore,

$$s_{i,l} = \begin{cases} f(x_{i,l}) - U_{l-1}[f(x_{i,l})], & \text{for } x_{i,l} \in \mathcal{D}_l, \\ 0, & \text{for } x_{i,l} \in \mathcal{H}_l \setminus \mathcal{D}_l. \end{cases} \tag{2.45}$$

It follows from Eq. (2.45) that the hierarchical surplus of any given node characterises the total contribution of that node to the interpolation. Consequently, a single, and unique, basis function is associated with every node. One can therefore write the delta interpolation operator as

$$\Delta U_j[f(x)] = \sum_{x_{i,j} \in \mathcal{D}_j} s_{i,j}\phi_{i,j}(x). \tag{2.46}$$

This allows us to write the interpolation operator (2.38) in the hierarchical form

$$U_l[f(x)] = \sum_{j=1}^{l} \Delta U_j[f(x)] = \sum_{j=1}^{l} \sum_{x_{i,j} \in \mathcal{D}_j} s_{i,j}\phi_{i,j}(x). \tag{2.47}$$

Hierarchical interpolation, as presented here, can also be interpreted as iteratively interpolating the remainder from the previous level interpolation.

A specific example may make the process of interpolating in a hierarchical manner clearer. Translating the above general discussion to the specific case of Lagrange interpolation on Chebyshev nodes, defined in Eq. (2.28), leads to $U_l[f(x)] = L_{n_l}(x)$ and $s_{i,l} = f(x_{i,l}) - L_{n_{l-1}}(x)$. The basis functions are the Lagrange basis polynomials, $\theta_{i,l}(x)$ and Figure 2.2 shows the $\theta_{i,l}(x)$ that are associated with

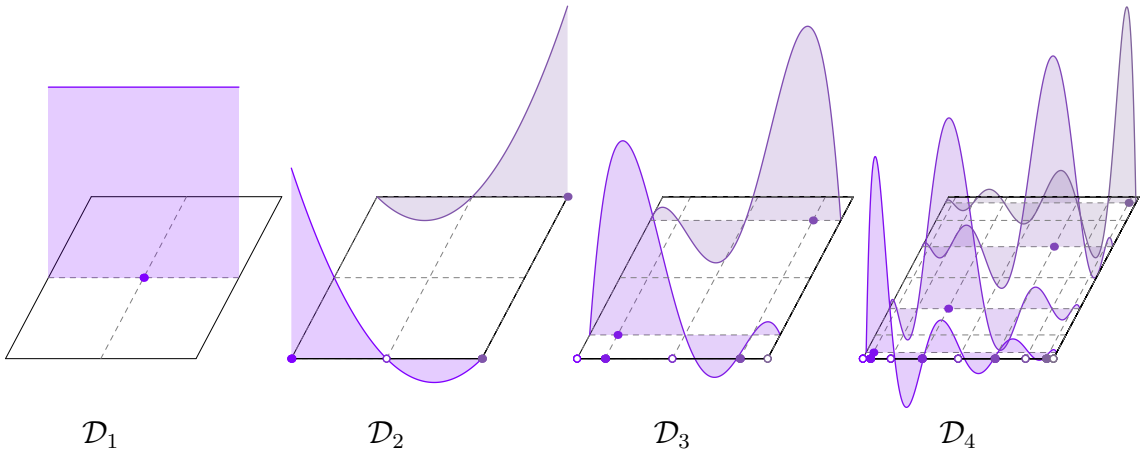


Figure 2.2: Lagrange basis polynomials

a delta grid for levels $j = 1, 2, 3$ and 4.

Figure 2.2 also shows the nodes associated with each delta grid. One can see that the first node to appear will be the node at the centre of the interval, 0.5 in this case. The polynomial associated with it is a constant function with magnitude 1, as seen in the left-most picture in Figure 2.2. The next level grid, $\mathcal{H}_2$ contains 3 nodes, the one in the centre and at the edges of the interval. Only the nodes at the edges of the interval are new to level $l = 2$, therefore $\mathcal{D}_2 = \{0, 1\}$. The second picture from the left in Figure 2.2 shows the basis polynomial associated with each of the two new nodes. These polynomials reach their maximum value of 1 at their associated nodes and are 0 at the node from the previous level, 0.5. Similarly, there are two nodes that are new to level 3, therefore $\mathcal{H}_3$ has five nodes. The basis polynomials of $x_{i,3}$ have a value of one at the nodes in $\mathcal{D}_3$ and are 0 at all the other nodes in $\mathcal{H}_3$, which include the nodes of $\mathcal{H}_2$ and $\mathcal{H}_1$.

Hierarchical interpolation proceeds as follows:

- sample $f(x)$ at the single node in $\mathcal{D}_1$ and use that value as a first level interpolation of $f(x)$;
- sample the function at the two nodes that are in $\mathcal{D}_2$, i.e. which are new to level $l = 2$;
- calculate the difference between the function value at the nodes in $\mathcal{D}_2$ and the value of the interpolation that was constructed with the node from $\mathcal{H}_1$ (the hierarchical surpluses);
- construct an interpolation on the hierarchical surpluses using the basis functions associated with the nodes at which they were sampled; and
- repeat the last three steps for all the remaining levels.

2.1.5.3 The representation operator

In order to generalise and summarise the above discussion, let us introduce a generic representation operator, $W[f(x)]$ and define it as

$$W[f(x)] = \sum_{i=1}^{n} \phi_i(x)\Lambda_i[f(x)]. \tag{2.48}$$

The operator Λ_i is defined as

$$\Lambda_i[f(x)] = \langle \zeta_i(x), f(x) \rangle = \int_{\mathcal{I}^d} \zeta_i(x) f(x) dx \tag{2.49}$$

with $\zeta_i(x)$ a given, suitable test function. This definition of the representation is a generalisation of Eqs. (2.11) and (2.21) and $f_n(x) = W[f(x)]$. In the case of interpolation it has already been shown that $\zeta_i(x) = \delta(x - x_i)$ and in the case of approximation $\zeta_i(x) = \phi_i(x)$. The representation operator as it is introduced here will be used in Section 2.2 where function representation in multiple dimensions will be discussed.

2.2 Function Representation in Multiple Dimensions

2.2.1 Problem description

Let us now consider the problem of representing a real-valued, multi-variate function, $f(\mathbf{x})$, that depends on the d input variables $\mathbf{x} = (x_1, \ldots, x_d)$. We assume that $f(\mathbf{x})$ is defined and continuous on the rectangular domain $\Omega^d = [a_i, b_i]^d \subset \mathbb{R}^d$. Since Ω^d is a compact subspace of $\mathbb{R}^d$, $f(\mathbf{x}) \in C(\Omega^d)$, which is the space of all functions that are continuous over Ω^d. This allows us to use the completeness property of the space of all functions that are continuous over a given compact subspace of $\mathbb{R}^d$ to write the multi-dimensional analogue of Eq. (2.1) as

$$f(\mathbf{x}) = \sum_{\mathbf{i} \in \mathcal{P}_\infty} c_{\mathbf{i}} \psi_{\mathbf{i}}(\mathbf{x}), \tag{2.50}$$

where $\psi_{\mathbf{i}}(\mathbf{x})$ forms a complete basis of $C(\Omega^d)$, and $c_{\mathbf{i}}$ are once again some coefficients that need to be determined. The sum is now over a vector of indices, referred to as a multi-index vector, $\mathbf{i} = (i_1, \ldots, i_d) \in \mathbb{N}^d$, which comes from the set of index vectors $\mathcal{P}_\infty = \{\mathbf{i} : 1 \leq i_k \leq \infty \text{ for } k = 1, \ldots, d\} = \mathbb{N}^d$. New index vectors are constructed by increasing any of the i_k. The enumerably infinite series of multi-index vectors, $\mathcal{P}_\infty$, can be truncated to some finite subset $\mathcal{P} \in \mathcal{P}_\infty$ in order

to obtain the multi-dimensional analogue of Eq. (2.2) (Griebel, 2006),

$$f(\mathbf{x}) \simeq f_N(\mathbf{x}; \mathbf{c_i}) = \sum_{\mathbf{i} \in \mathcal{P}} c_{\mathbf{i}} \psi_{\mathbf{i}}(\mathbf{x}), \tag{2.51}$$

where N is the total number of samples used to construct the multi-dimensional representation and is equal to the cardinality of the set $\mathcal{P}$. This truncation is essential to making a representation practical for everyday use, therefore choosing $\mathcal{P}$ will be central to this work and will be discussed in more detail in later sections.

For convenience, from here on forward Ω^d has been mapped to the unit hypercube and used as such $\Omega^d \to \mathcal{I}^d = [0,1]^d$.

2.2.2 The representation operator

The method used in this work to extend the representation to d dimensions was to construct a tensor product of the one-dimensional representation operator defined in Eq. (2.48) (Smoljak, 1963),

$$W^d[f(\mathbf{x})] = (W_1 \otimes \cdots \otimes W_d)\,[f(\mathbf{x})] \tag{2.52}$$

where the symbol $\otimes$ denotes the tensor product. The notation $\bigotimes_{k=1}^{d}$, with the associated meaning of performing a tensor product of a one-dimensional operator d times, is introduced for the sake of compactness, which allows us to write:

$$\begin{aligned} W^d[f(\mathbf{x})] &= \bigotimes_{k=1}^{d} W_k[f(\mathbf{x})] \\ &= \bigotimes_{k=1}^{d} \left(\sum_{i_k=1}^{n_k} \phi_{i_k}(x_k) \Lambda_{i_k} \right) [f(\mathbf{x})]. \end{aligned} \tag{2.53}$$

The integration operator, Λ_{i_k}, that acts on functions of x_k as defined in Eq. (2.49) does not commute with $\phi_{i_k}(x_k)$, but it does commute with $\phi_{i_j}(x_j)$ for all $j \neq k$, since Λ_k only acts on functions of x_k. Similarly, integration is self-commutative, thus Λ_k commutes with Λ_j. These two properties, along with the linearity of the representation operator and the properties of the tensor product, is then used to obtain:

$$W^d[f(\mathbf{x})] = \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \left(\phi_{i_1}(x_1) \otimes \cdots \otimes \phi_{i_d}(x_d)\right) \left(\Lambda_{i_1} \otimes \cdots \otimes \Lambda_{i_d}\right) [f(\mathbf{x})]. \tag{2.54}$$

To simplify this equation, let us introduce the multi-dimensional basis functions $\psi_{\mathbf{i}}(\mathbf{x})$, which are constructed as a tensor product of orthogonal one-dimensional basis functions,

$$\psi_{\mathbf{i}}(\mathbf{x}) = \phi_{i_1}(x_1) \otimes \cdots \otimes \phi_{i_d}(x_d). \tag{2.55}$$

Since both the Legendre polynomials and the Lagrange basis polynomials, which are the functions that were used as one-dimensional basis functions in this work, are real valued scalar functions, the tensor product is replaced with the ordinary product:

$$\psi_{\mathbf{i}}(\mathbf{x}) = \prod_{k=1}^{d} \phi_{i_k}(x_k). \tag{2.56}$$

Similarly, we introduce the multi-dimensional test functions $\xi_{\mathbf{i}}(\mathbf{x})$ that are constructed as

$$\xi_{\mathbf{i}}(\mathbf{x}) = \zeta_{i_1}(x_1) \otimes \cdots \otimes \zeta_{i_d}(x_d). \tag{2.57}$$

In this work only real-valued scalar functions will be used as test functions, which again allows us to replace the tensor product with the ordinary product

$$\xi_{\mathbf{i}}(\mathbf{x}) = \prod_{k=1}^{d} \zeta_{i_k}(x_k). \tag{2.58}$$

Since the Λ_{i_k} operate on different variables, x_k, they are commutative and the following holds:

$$\begin{aligned} \Lambda_{i_1} \otimes \cdots \otimes \Lambda_{i_d}[f(\mathbf{x})] &= \int_{\mathcal{I}^d} \zeta_{i_1}(x_1) \cdots \zeta_{i_d}(x_d) f(x_1, \ldots, x_d) dx_1 \cdots dx_d \\ &= \int_{\mathcal{I}^d} \xi_{\mathbf{i}}(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \end{aligned} \tag{2.59}$$

which leads to

$$\begin{aligned} W^d[f(\mathbf{x})] &= \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \psi_{\mathbf{i}}(\mathbf{x}) \int_{\mathcal{I}^d} \xi_{\mathbf{i}}(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \\ &= \sum_{\mathbf{i} \in \mathcal{P}} \psi_{\mathbf{i}}(\mathbf{x}) \int_{\mathcal{I}^d} \xi_{\mathbf{i}}(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \end{aligned} \tag{2.60}$$

where $\mathcal{P}$ is an arbitrary set of indices which will be defined later. The discussion of multi-dimensional interpolation will now follow.

2.2.3 Interpolation

In the case of interpolation, we introduce the interpolation operator, $U[f(x)]$ and choose $\zeta_i(x) = \delta(x - x_i)$. It has been shown in previous sections that this test function leads to interpolation and

in this case $W[f(x)] = U[f(x)]$. Taking the product of the ζ_{i_k} in Eq. (2.60), we obtain

$$\xi_{\mathbf{i}}(\mathbf{x}) = \prod_{k=1}^{d} \zeta_{i_k}(x_k) = \prod_{k=1}^{d} \delta(x_k - x_{i_k}) = \delta(\mathbf{x} - \mathbf{x}_i). \tag{2.61}$$

In Cartesian coordinates $\delta(\mathbf{x} - \mathbf{x}_i)$ corresponds to the traditional interpretation, i.e. the Dirac delta is applied coordinate-wise (Boas, 2006). The multi-dimensional interpolation operator can then be written in the form

$$\begin{aligned} U^d[f(x)] &= \sum_{\mathbf{i}\in\mathcal{P}} \psi_{\mathbf{i}}(\mathbf{x}) \int_{\mathcal{I}^d} \delta(\mathbf{x} - \mathbf{x}_{\mathbf{i}}) f(\mathbf{x}) d\mathbf{x} \\ &= \sum_{\mathbf{i}\in\mathcal{P}} \psi_{\mathbf{i}}(\mathbf{x}) f(\mathbf{x}_{\mathbf{i}}), \end{aligned} \tag{2.62}$$

in other words $c_{\mathbf{i}} = f(\mathbf{x}_{\mathbf{i}})$ in Eq. (2.51). Note that the set of indices $\mathcal{P}$ has not yet been fixed and is still in some sense arbitrary.

2.2.4 Approximation

In the case of approximation, we introduce the approximation operator, $Y[f(x)]$. It has been shown before that $\zeta_i(x) = \phi_i(x)$ leads to approximation and in this case $W[f(x)] = Y[f(x)]$. Taking the tensor product of the ζ_i in Eq. (2.60), we obtain

$$\xi_{\mathbf{i}}(\mathbf{x}) = \prod_{k=1}^{d} \phi_{i_k}(x_k) = \psi_{\mathbf{i}}(\mathbf{x}). \tag{2.63}$$

The multi-dimensional approximation operator can then be written in the form

$$Y^d[f(x)] = \sum_{\mathbf{i}\in\mathcal{P}} \psi_{\mathbf{i}}(\mathbf{x}) \int_{\mathcal{I}^d} \psi_{\mathbf{i}}(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}. \tag{2.64}$$

The enumerable, infinite set $\mathcal{P}_\infty$ can be limited in several ways to obtain a finite set of index vectors, $\mathcal{P}$. In this work, $\mathcal{P}_\infty$ is limited by placing bounds on the vector norms e.g. (An and Owen, 2001),

$$\begin{aligned} \mathcal{B}_0 &= \{\mathbf{i} \in \mathcal{P}_\infty : ||\mathbf{i}||_0 \leq B_0\} \\ \mathcal{B}_1 &= \{\mathbf{i} \in \mathcal{P}_\infty : ||\mathbf{i}||_1 \leq B_1\} \\ \mathcal{B}_\infty &= \{\mathbf{i} \in \mathcal{P}_\infty : ||\mathbf{i}||_\infty \leq B_\infty\} \end{aligned}$$

where $||\mathbf{i}||_0$ is the number of $i_k > 1$, $||\mathbf{i}||_1 = \sum_{k=1}^{d} i_k$ is the total order, $||\mathbf{i}||_\infty$ is the maximum one-dimensional order and the constants B_0, B_1, B_∞ are appropriate pre-determined bounds. The intersection of the sets $\mathcal{B}_0$, $\mathcal{B}_1$ and $\mathcal{B}_\infty$ are used to generate a set of indices of the basis functions

when applying approximation,

$$\mathcal{P} = \mathcal{B}_{0,1,\infty} = \mathcal{B}_0 \cap \mathcal{B}_1 \cap \mathcal{B}_\infty. \tag{2.65}$$

The $c_{\mathbf{i}}$ in Eq. (2.51) need to be determined by calculating a multi-dimensional integral,

$$c_{\mathbf{i}} = \int_{\mathcal{I}^d} \psi_{\mathbf{i}}(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}, \tag{2.66}$$

assuming orthonormal basis functions. As in the one-dimensional case the $c_{\mathbf{i}}$ need to be approximated numerically by applying a quadrature, but this time in multiple dimensions. The tensor product construction is used again to create a multi-dimensional quadrature.

2.2.5 Tensor product quadrature

The multi-dimensional quadrature operator, $Q_{\mathbf{l}}^d[f(\mathbf{x})]$, can be introduced as a tensor product of the one-dimensional quadrature rules, which proceeds as (Gerstner and Griebel, 1998)

$$\begin{aligned} Q_{\mathbf{l}}^d[f(\mathbf{x})] &= \left(Q_{l_1} \otimes \cdots \otimes Q_{l_d}\right)[f(\mathbf{x})] \\ &= \sum_{i_d=1}^{n_{l_d}} \cdots \sum_{i_1=1}^{n_{l_1}} w_{i_1,l_1} \cdots w_{i_d,l_d} f\left(x_{i_1,l_d}, \ldots, x_{i_d,l_d}\right). \end{aligned} \tag{2.67}$$

Let us define the multi-dimensional quadrature weights

$$w_{\mathbf{i},\mathbf{l}} = w_{i_1,l_1} \cdots w_{i_d,l_d} = \prod_{k=1}^{d} w_{i_k,l_k} \tag{2.68}$$

which allows us to write Eq. (2.67) in the form

$$Q_{\mathbf{l}}^d[f(\mathbf{x})] = \sum_{\mathbf{i} \in \mathcal{P}_{\mathrm{TP}}} w_{\mathbf{i},\mathbf{l}} f(x_{\mathbf{i},\mathbf{l}}), \tag{2.69}$$

where $\mathcal{P}_{\mathrm{TP}} = \{\mathbf{i} \in \mathbb{N}^d : 1 \leq i_k \leq n_k \text{ for } k = 1, \ldots, d\}$ defines the set of indices that identify the samples that were used for the quadrature.

2.2.6 Cartesian product grids

We recall that the set of Chebyshev nodes, $\mathcal{H}_l$, forms the one-dimensional Chebyshev grid of accuracy level l. To extend that grid to multiple dimensions, we write the *Cartesian product* of one-dimensional Chebyshev grids:

$$\mathcal{H}_{\mathbf{l}}^d = \mathcal{H}_{l_1} \times \cdots \times \mathcal{H}_{l_d} \tag{2.70}$$

where $\mathbf{l} = (l_1, \ldots, l_d)$ is a vector of the one-dimensional grid levels and $\mathcal{S}_\ell$ is the set of index vectors that define that maximum level of every dimension,

$$\mathcal{S}_\ell = \{\mathbf{l} : l_k \leq \ell_k,\ \ell_k \in \mathbb{N},\ \text{for } k = 1, \ldots, d\}.$$

If one sets all ℓ_k equal to ℓ, one obtains an isotropic Cartesian grid. One can uniquely identify each sample using the two indices $\mathbf{i}$ and $\mathbf{l}$, as in Section 2.1.4,

$$x_{\mathbf{i},\mathbf{l}} \in \mathcal{H}_\mathbf{l} \iff \mathbf{i} \in \mathcal{P}_{\text{TP}} \text{ and } \mathbf{l} \in \mathcal{S}_\ell.$$

This defines the indices $\mathbf{i}$ in a standard tensor product construction, where the one-dimensional rules that underpin this tensor product can have a different level in each dimension k. The sets $\mathcal{P}_{\text{TP}}$ and $\mathcal{S}_\ell$ together define the samples that are needed for tensor product interpolation or quadrature on Chebyshev nodes.

2.3 Sparse Grids

2.3.1 Isotropic case

Tensor product grids have many features that make them attractive for function representation, but they have the draw-back of the curse of dimension, which limits their use for many practical applications, including cross section representation. Sparse grid methods use combinations of low-order tensor product discretisation grids to mitigate the curse of dimension and as such a sparse grid is a subset of the equivalent full grid, as can be seen in Figure 1.1. Smoljak (1963) proposed this method for combining tensor products of one-dimensional interpolation or quadrature rules to exploit any smoothness in the underlying function.

Let us introduce the notion of a *valid index* that comes from an *admissible set*. The set from which the elements in a tensor product construction may be chosen is called the admissible set of that construction. The admissible set may contain different kinds of elements for different types of construction. As an example, for interpolation and quadrature, the admissible set may contain the indices of grid points at which samples are taken, such as $\mathcal{P}_{\text{TP}}$ above, or the levels that may be used, such as $\mathcal{S}_\ell$ above.

Sparse grids will be applied to quadrature and interpolation, and when the term admissible set is used in this section, it will refer to some set from which the levels of the underlying tensor product grids may be chosen. To construct a sparse grid by the Smoljak construction, we define the admissible set of levels as

$$\mathcal{S}_\Delta = \left\{ \mathbf{l} \in \mathbb{N}^d : q \geq \sum_{k=1}^{d} (l_k - 1)\,,\ l_k \geq 1 \text{ for } 1 \leq k \leq d \right\} \tag{2.71}$$

where q is called the level of the sparse grid. Note that there are now four different types of levels: the level of the one-dimensional grid, l, the level of a node in the one-dimensional grid, j, the vector of levels in a tensor product grid, $\mathbf{l}$, and now the sparse grid level, which is a scalar quantity. Close inspection of Eq. (2.71) reveals that, although the one-dimensional level starts at one, the sparse grid level starts at zero. This admissible set of levels, $\mathcal{S}_\Delta$, may then be used to construct either a sparse grid interpolation rule or a sparse grid quadrature rule.

Let us start by producing a sparse grid quadrature rule. Given the definition of ΔQ_l in Eq. (2.35) and the definition of tensor product quadrature in Eq. (2.67), the Smoljak construction that produces a sparse grid quadrature rule is

$$\mathbb{Q}_q^d[f(\mathbf{x})] = \sum_{\mathbf{l}\in\mathcal{S}_\Delta} \left(\Delta Q_{l_1} \otimes \cdots \otimes \Delta Q_{l_d}\right)[f(\mathbf{x})]. \tag{2.72}$$

where $\mathbb{Q}_q^d[f(\mathbf{x})]$ denotes the sparse grid quadrature operator. This sparse grid formulation may also be written in the level-iterative way:

$$\begin{aligned}\mathbb{Q}_q^d[f(\mathbf{x})] &= \sum_{m=1}^{q} \sum_{||\mathbf{l}||_1=m+d} \left(\Delta Q_{l_1} \otimes \cdots \otimes \Delta Q_{l_d}\right)[f(\mathbf{x})] \\ &= \sum_{m=1}^{q} \sum_{||\mathbf{l}||_1=m+d} \Delta Q_{\mathbf{l}}^d[f(\mathbf{x})].\end{aligned} \tag{2.73}$$

It is also possible to write Eq. (2.72) in a combinatorial form by choosing a different admissible set of levels and using the definition of Q_l in Eq. (2.33). Let us define the admissible set of levels for the combinatorial construction as

$$\mathcal{S}_{\text{com}} = \left\{\mathbf{l}\in\mathbb{N}^d,\ l_k \geq 1\ \forall k\in\mathbb{N} : q-d+1 \leq \sum_{k=1}^{d}(l_k-1) \leq q\right\}. \tag{2.74}$$

Then the combinatorial formula for sparse grid quadrature becomes (Wasilkowski and Wozniakowski, 1995):

$$\mathbb{Q}_q^d[f(\mathbf{x})] = \sum_{\mathbf{l}\in\mathcal{S}_{\text{com}}} (-1)^{q+d-||\mathbf{l}||_1} \begin{pmatrix} d-1 \\ q+d-||\mathbf{l}||_1 \end{pmatrix} \left(Q_{l_1} \otimes \cdots \otimes Q_{l_d}\right)[f(\mathbf{x})]. \tag{2.75}$$

This combinatorial form is equivalent to the level-iterative form in Eq. (2.73).

In an similar manner, given the definition of ΔU_l in Eq. (2.39) and the definition of tensor product interpolation in Eq. (2.62), sparse grid interpolation can be constructed by Smolyak's method,

$$\mathbb{U}_q^d[f(\mathbf{x})] = \sum_{\mathbf{l}\in\mathcal{S}_\Delta} \left(\Delta U_{l_1} \otimes \cdots \otimes \Delta U_{l_d}\right)[f(\mathbf{x})], \tag{2.76}$$

or the level-iterative form

$$\begin{aligned}\mathbb{U}_q^d[f(\mathbf{x})] &= \sum_{m=1}^{q} \sum_{||\mathbf{l}||_1=m+d} \left(\Delta U_{l_1} \otimes \cdots \otimes \Delta U_{l_d}\right)[f(\mathbf{x})] \\ &= \sum_{m=1}^{q} \sum_{||\mathbf{l}||_1=m+d} \Delta U_{\mathbf{l}}^d[f(\mathbf{x})].\end{aligned} \tag{2.77}$$

Given the definition of U_l in Eq. (2.38) and the admissible set (2.74), sparse grid interpolation may be constructed by the combinatorial method,

$$\mathbb{U}_q^d[f(\mathbf{x})] = \sum_{\mathbf{l}\in\mathcal{S}_{\text{com}}} (-1)^{q+d-||\mathbf{l}||_1} \begin{pmatrix} d-1 \\ q+d-||\mathbf{l}||_1 \end{pmatrix} \left(U_{l_1} \otimes \cdots \otimes U_{l_d}\right)[f(\mathbf{x})]. \tag{2.78}$$

The original formulation of Smoljak's method was in terms of delta operators, such as Eq. (2.72) and Eq. (2.76). The combinatorial form, Eq. (2.75) and Eq. (2.78), was developed later, because it is conceptually easier to deal with and more straightforward to implement. The implementation makes it difficult, however, to perform hierarchical quadrature or interpolation on a nested grid without re-sampling all nodes at a given level. The implementation in this work is based on the original formulation, because all constructions were done hierarchically and the function that we deal with is expensive to sample, thus it is imperative that the same node not be sampled twice, even if it is used repeatedly.

The nodes that result from Smoljak's algorithm can be described as a union of nodes from different Cartesian product grids. We denote the set of nodes that form a sparse grid of level q as $\mathbb{H}_q^d$. If the sets $\mathcal{H}_{l_k}$ are nested then the sparse grid built from their union is also nested, i.e $\mathbb{H}_q^d \subset \mathbb{H}_{q+1}^d$, therefore the set of points that constitute the "sparse grid" becomes

$$\mathbb{H}_q^d = \bigcup_{\mathbf{l}\in\mathcal{S}_{\text{nest}}} \mathcal{H}_{\mathbf{l}}^d = \bigcup_{\mathbf{l}\in\mathcal{S}_{\text{nest}}} \left(\mathcal{H}_{l_1} \times \cdots \times \mathcal{H}_{l_d}\right) \tag{2.79}$$

where

$$\mathcal{S}_{\text{nest}} = \left\{ \mathbf{l} \in \mathbb{N}^d,\ l_k \geq 1 \text{ for } k \in \mathbb{N} : q = \sum_{k=1}^{d} (l_k - 1) \right\} \tag{2.80}$$

is the set of all one-dimensional levels that form part of the highest sparse grid level.

The discussions on delta grids remain valid for sparse grids. The delta sets $\mathcal{D}_l$ can be extended to multiple dimensions in a similar way, $\mathcal{D}_{\mathbf{l}}^d = \mathcal{D}_{l_1} \otimes \cdots \otimes \mathcal{D}_{l_d}$. In multiple dimensions, the union of the delta sets must still be equal to the full grid,

$$\mathcal{H}_{\mathbf{l}}^d = \bigcup_{\substack{j_k \leq l_k \\ 1\leq k\leq d}} \mathcal{D}_{\mathbf{j}}^d. \tag{2.81}$$

$\mathcal{D}_{\mathbf{l}}^d$ still represents all the nodes (in $\mathcal{I}^d = [0,1]^d$) that are new to $\mathcal{H}_{\mathbf{l}}^d$. Sparse grid quadrature and interpolation are both performed on the same set of samples, which form the grid of the "sparse grid",

$$\mathbb{H}_q^d = \bigcup_{\mathbf{l}\in\mathcal{S}_{\text{com}}} \mathcal{H}_{\mathbf{l}}^d = \bigcup_{\mathbf{l}\in\mathcal{S}_{\text{com}}} \left(\mathcal{H}_{l_1} \times \cdots \times \mathcal{H}_{l_d}\right). \tag{2.82}$$

2.3.2 Anisotropic case

Isotropic sparse grids are efficient for constructing representations of functions that are equally dependent on all the input parameters, or of which the dependence is unknown. If there is a priori information available that the function being represented depends more strongly on some of the input parameters than on others, one can use an anisotropic sparse grid to account for this.

To extend the theory of isotropic sparse grids to anisotropic grids, we introduce a weight factor $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}^d$ and define

$$\alpha_{\min} = \min_{1\le k\le d} \alpha_k$$

where each α_k is a real-valued number larger than zero. In this work only integer values were used for α_k. The set of admissible indices that lead to the anisotropic Smoljak construction is (Nobile, Tempone, and Webster, 2008b)

$$\mathcal{S}_{\boldsymbol{\alpha}} := \left\{ \mathbf{l} \in \mathbb{N}^d,\ l_k \ge 1 \text{ for } k \in \mathbb{N} : q\alpha_{\min} \ge \sum_{k=1}^{d} (l_k - 1)\,\alpha_k \right\}. \tag{2.83}$$

The difference between an isotropic and an anisotropic sparse grid is illustrated in Figure 2.3. One can see from Figure 2.3 that the effect of using the admissible set $\mathcal{S}_{\boldsymbol{\alpha}}$ is not to add extra nodes in a certain dimension, but to remove nodes from the isotropic grid of the same level, preferentially but not exclusively from the dimensions that have a higher value of α than the dimension that carries a heavier weight.

Using set $\mathcal{S}_{\boldsymbol{\alpha}}$ as the admissible set yields the Smolyak constructions for quadrature and interpolation, respectively as

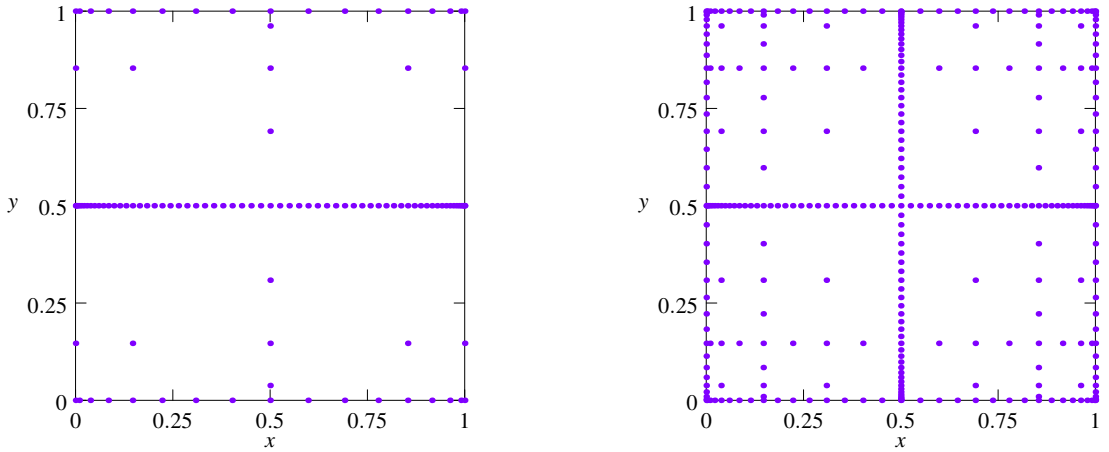
$$\mathbb{Q}_q^d[f(\mathbf{x})] = \sum_{\mathbf{l}\in\mathcal{S}_{\boldsymbol{\alpha}}} \left(\Delta Q_{l_1} \otimes \cdots \otimes \Delta Q_{l_d}\right)[f(\mathbf{x})] \tag{2.84}$$

and

$$\mathbb{U}_q^d[f(\mathbf{x})] = \sum_{\mathbf{l}\in\mathcal{S}_{\boldsymbol{\alpha}}} \left(\Delta U_{l_1} \otimes \cdots \otimes \Delta U_{l_d}\right)[f(\mathbf{x})]. \tag{2.85}$$

2.3.3 Convergence rate

It has already been mentioned that sparse grid methods make use of and rely on the smoothness of the function for which a representation is sought. Specifically, the rate at which the error converges



(a) Sparse grid with anisotropy vector $\boldsymbol{\alpha} = (1, 2)$ **(b) Isotropic sparse grid**

Figure 2.3: Comparison between a two-dimensional isotropic sparse grid and an anisotropic sparse grid, both of level six

depends on whether the function has bounded mixed derivatives. This means we consider functions from the class $\mathcal{W}_r^d$ where (Gerstner and Griebel, 1998; Gerstner, 2007)

$$\mathcal{W}_r^d := \left\{ \Omega \to \mathbb{R}, \left\| \frac{\partial_1^{||\mathbf{s}||_1} f}{\partial x_1^{s_1} \cdots \partial x_d^{s_d}} \right\|_\infty < \infty, s_i \leq r \right\}. \tag{2.86}$$

Under this smoothness condition it can be shown that if one starts with a one-dimensional representation Q_l that has an error bound $|f(x) - Q_l| = \mathcal{O}(n_l^{-r})$, a condition that holds for both quadrature and interpolation on Chebyshev nodes, it follows that (Novak and Ritter, 1996; Barthelmann, Novak, and Ritter, 2000; Gerstner, 2007)

$$\left| f(\mathbf{x}) - W_\ell^d[f(\mathbf{x})] \right| = \mathcal{O}(N^{-r} (\log N)^{(d-1)(r+1)}), \tag{2.87}$$

where $\mathcal{O}$ is the Landau symbol that indicates the rate of growth of a function. An alternative estimate for the decay of the error was derived by Nobile, Tempone, and Webster (2008a), specifically for isotropic sparse grid methods based on a Chebyshev grid:

$$\left| f(\mathbf{x}) - W_\ell^d[f(\mathbf{x})] \right| = \mathcal{O}(N^{-\sigma/(\log(2d))}), \tag{2.88}$$

where $\sigma = r$ for $f(\mathbf{x}) \in \mathcal{W}_r^d$. This gives us theoretical convergence rates to which we can compare the results that will be obtained.

2.4 Model Optimisation and Error Control

It is important to have information about the accuracy of the representation, i.e. how closely $f_N(\mathbf{x})$ approximates $f(\mathbf{x})$. Ideally, any representation method would have as its output the coefficients $c_{\mathbf{i}}$, as well as an estimate of the accuracy. Sparse grid methods offer ways to both quantify the accuracy and optimise the representation that is produced in terms of storage requirements and reconstruction time.

There are two possible sources of error in any representation: truncation error and estimation error. Truncation error is the error introduced into the representation by discarding terms in the representation, either by the choice of the accuracy level (and therefore, by implication, N) or by discarding individual terms deemed to make a negligible contribution. Estimation error is introduced by inaccuracies in estimating the representation coefficients. Both approximation and interpolation may incur truncation error. In addition to truncation error, approximation may also incur estimation error, due to the limited accuracy of the quadrature method used. The cases of approximation and interpolation will be discussed separately, because they have different sources of inaccuracy and because the representation error is quantified in different ways. The total error of the representation needs to be quantified in a consistent manner and the most suitable metric by which to quantify the inaccuracies in an approximation is by the least square or L_2 norm, while the most natural metric in the case of interpolation is the maximum or L_∞ norm.

It is not always necessary, or desirable, to retain all N terms in Eq. (2.51). Some terms may make a negligible contribution to the representation. Discarding these terms may improve reconstruction time as well as reduce the amount of memory needed to store the representation. The amount of memory occupied by the representation will be an important consideration once multi-dimensional representation is used. It is our objective to find a good balance between discarding unnecessary terms and minimising the representation error. This process will be referred to as *model reduction* or *model optimisation*.

The magnitude of a coefficient $c_{\mathbf{i}}$ from Eq. (2.51) can give some indication of the contribution that the associated term, $c_{\mathbf{i}}\psi_{\mathbf{i}}(\mathbf{x})$, makes to the representation, either in the least square sense as in approximation or in the maximum sense as in interpolation. This information can then be used to judge what effect discarding term $\mathbf{i}$ would have on the truncation error and thereby apply effective *error control*.

2.4.1 Error control and model reduction in approximation

2.4.1.1 Analysis of variance

One way of estimating the contribution of each term to the representation is based on functional Analysis of Variance (ANOVA), as discussed by An and Owen (2001) and Lemieux and Owen (2002).

This approach yields the magnitude of any term's contribution in the L_2 norm and the absolute error in this norm is $\epsilon_2 = ||f(\mathbf{x}) - f_N(\mathbf{x})||_2$. The elements of functional ANOVA that are pertinent to error control and model reduction in approximation will now be covered.

Let us define, for a given function $f(\mathbf{x})$, the expectation value

$$E[f(\mathbf{x})] = \int_{\mathcal{I}^d} f(\mathbf{x}) d\mathbf{x} \tag{2.89}$$

and the variance of the function,

$$\mathrm{Var}[f(\mathbf{x})] = E\left[(f(\mathbf{x}) - E[f(\mathbf{x})])^2\right]. \tag{2.90}$$

Since the first Legendre polynomial $\psi_{\mathbf{1}}(\mathbf{x}) = \prod_{k=1}^{d} \phi_{1_k}(x_k) = 1$, it follows that

$$c_{\mathbf{1}} = \int_{\mathcal{I}^d} \psi_{\mathbf{1}}(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} = E[f(\mathbf{x})]. \tag{2.91}$$

This allows us to define the variance of $f(\mathbf{x})$ as

$$\begin{aligned} \mathrm{Var}[f(\mathbf{x})] &= \int_{\mathcal{I}^d} f^2(\mathbf{x}) d\mathbf{x} - \left(\int_{\mathcal{I}^d} f(\mathbf{x}) d\mathbf{x} \right)^2 \\ &= \int_{\mathcal{I}^d} f^2(\mathbf{x}) d\mathbf{x} - c_{\mathbf{1}}^2. \end{aligned} \tag{2.92}$$

As long as the basis functions are orthonormal and the domain of integration is $\mathcal{I}^d$ (which is the reason why the Chebyshev nodes were rescaled to $[0, 1]$), the squares of the coefficients in Eq. (2.51), $c_{\mathbf{i}}^2$, sum to the variance of $f(\mathbf{x})$ (Lemieux and Owen, 2002):

$$\mathrm{Var}[f(\mathbf{x})] = \sum_{\substack{\mathbf{i} \in \mathcal{P}_\infty \\ \mathbf{i} \neq \mathbf{1}}} c_{\mathbf{i}}^2. \tag{2.93}$$

Note that the constant term does not contribute to the function variance. As one can see from Eq. (2.93), the orthogonality of the basis functions implies that the contribution of each term $\mathbf{i}$ to the function variance is independent from that of all the other terms.

Dividing both sides of Eq. (2.93) by the variance gives the fraction of the function variance that results from of each term. These fractions must then sum to 1 if the set $\mathcal{P}_\infty$ is used and all coefficients are exact, or free of estimation error. After limiting the set to $\mathcal{P}$, the contribution of each term in

$\mathcal{P}$ to the variance must still sum to a good approximation of the function variance,

$$\sum_{\substack{\mathbf{i}\in\mathcal{P}\\ \mathbf{i}\neq\mathbf{1}}} \frac{c_{\mathbf{i}}^2}{\mathrm{Var}[f(x)]} \approx 1. \tag{2.94}$$

If the fraction of any term in Eq. (2.94) is sufficiently smaller than 1, it follows immediately that its contribution to the total variance of the function is small and that term can safely be neglected from the representation. The user may control the truncation error by choosing the minimum contribution a term needs to make in order to be included in the representation. The variance of the function must be known in order to determine the truncation error, which would be the difference between the variance and the sum of the squared coefficients which correspond to terms from $\mathcal{P}$. In practice the variance is estimated independently by solving the integral in Eq. (2.92) numerically.

A rough measure of the error in estimating the coefficients can be obtained from the difference between the estimated value of the coefficient at the current quadrature level and the estimated value at the previous level. This measure is conservative and can be used to exclude terms that would introduce a large estimation error into the representation. Neglecting insignificant or badly estimated terms that do not contribute significantly allows us to optimise the representation in terms of storage requirements and reconstruction time. More details of how the terms that are used in the optimised approximation are chosen, and the methods used to identify badly estimated terms, may be found in (Živanović and Bokov, 2010).

2.4.1.2 High dimensional model representation

The theory of High Dimensional Model Representation (HDMR) indicates that a function of d variables, $f(\mathbf{x}) = f(x_1, \ldots, x_d)$, can be decomposed into a hierarchical expansion with 2^d terms of increasing dimensions (Sobol', 2003; Rabitz et al., 1999):

$$\begin{aligned} f(\mathbf{x}) = f_0 &+ f_1(x_1) + \cdots + f_d(x_d) \\ &+ \sum_{k<m} f_{k,m}(x_k, x_m) \\ &\vdots \\ &+ f_{1,\ldots,d}(x_1, \ldots, x_d). \end{aligned} \tag{2.95}$$

Here f_0 is a constant term that does not depend on any input parameter. The functions of a single variable, $f_k(x_k)$, give the effect of each variable without any interactions with the other input variables. The functions of two variables, $f_{k,m}(x_k, x_m)$, describe the cooperative effect of two variables x_k and x_m, and terms with more variables show the interactions of increasing numbers of variables on which the function $f(\mathbf{x})$ depends simultaneously.

The motivation behind HDMR is the empirical observation that components of Eq. (2.95) for typical

real life problems are unlikely to have interaction of more the two input variables (Rabitz et al., 1999). In such a case, the expansion (2.95) can be truncated after the first two lines, yielding a representation that contains functions of two variables at most. Furthermore, HDMR is often used, in conjunction with other techniques, to analyse the internal structure (effective dimensionality, additivity, linearity, etc.) of complex or unknown 'black-box'-like dependencies.

Representations in the form of Eq. (2.95) are usually constructed via a constrained optimisation process similar to the mean weighted residuals method, discussed in Section 2.1.1 (see discussion in (Rabitz et al., 1999) for more details). A specific choice of constraints/conditions leads to a particular HDMR form. One of these forms, referred as ANOVA-HDMR, RS-HDMR, functional ANOVA or simply ANOVA in the literature, is of particular interest for this work.

ANOVA-HDMR can be constructed for square integrable functions by imposing certain conditions: that f_0 is constant and the integral of every term in Eq. (2.4) over any of its own variables is zero. These requirements can be shown to be equivalent to the condition of pair-wise orthogonality of the terms in Eq. (2.95) with respect to the inner product (2.4) (Sobol', 1993; Sobol', 2003; Rabitz et al., 1999).

The orthogonality of HDMR expansion terms leads to the property that the variance of the function $f(\mathbf{x})$ equals the sum of the variances of the terms in its decomposition,

$$\begin{aligned} \mathrm{Var}[f(\mathbf{x})] = \mathrm{Var}[f_0] + \mathrm{Var}[f_1(x_1)] + \cdots + \mathrm{Var}[f_d(x_d)] \\ + \sum_{k<m} \mathrm{Var}[f_{k,m}(x_k, x_m)] \\ \vdots \\ + \mathrm{Var}[f_{1,\ldots,d}(x_1, \ldots, x_d)], \end{aligned} \tag{2.96}$$

bearing in mind that $\mathrm{Var}[f_0] = 0$. Representations (2.51) and (2.95) can be shown to be equivalent by grouping like terms, (Lemieux and Owen, 2002; Bokov, 2009). Similarly, Eq. (2.93) is equivalent to Eq. (2.96), as a consequence of the same grouping of like terms. This allows us to repeat the analysis of the contribution that each term makes to the variance, which was done previously for the expansion in terms of basis functions and coefficients. The significance of any term in Eq. (2.96) can be quantified as the fraction of the variance that it carries and terms from Eq. (2.96) may be discarded based on this analysis. Information about the cross section model, such as which variables make a negligible contribution to the variance, and the maximum number of interacting variables (the effective dimension of the problem) may be extracted from the analysis. This allows us to better understand the structure of our model, based on information that was obtained by sampling the function.

2.4.2 Error control and model reduction in interpolation

Representation optimisation can also be performed in the case of interpolation. The cardinal functions used in this work are one at the node that the particular cardinal function is associated with and zero at every other node at that or previous levels ($\phi_j(x_i) = \delta_{ji}$). The maximum value that the cardinal function reaches is also approximately one. The coefficient of every term therefore indicates its contribution to the maximum relative error. More intuitively, the remainder of the previous level interpolation, and therefore the hierarchical surpluses in the current level interpolation, decrease as the level increases. It is therefore possible to estimate the absolute error in the L_∞ norm:

$$\epsilon_\infty = ||f(\mathbf{x}) - f_N(\mathbf{x})||_\infty = \max_{\mathbf{x}\in\mathcal{I}^d} |f(\mathbf{x}) - f_N(\mathbf{x})| \,. \tag{2.97}$$

If we use the interpolation at the current level as an estimator for $f(\mathbf{x})$, then

$$\epsilon_\infty \approx ||\mathbb{U}_q^d[f(\mathbf{x})] - \mathbb{U}_{q-1}^d[f(\mathbf{x})]||_\infty, \tag{2.98}$$

which can be estimated by the hierarchical surpluses from the delta set $\mathcal{D}_\mathbf{q}$ as

$$\epsilon_\infty^n \approx \max_{\mathbf{x_i}\in\mathcal{D}_q} |s_\mathbf{i}| \,. \tag{2.99}$$

Here ϵ_∞ is the true maximum error of interpolation and ϵ_∞^n is the maximum error as estimated from the hierarchical surpluses at the n points in $\mathcal{D}_q$. Here $\mathbf{i}$ is the multi-dimensional index of the node at sparse grid level q.

Since the maximum error is so closely associated with the hierarchical surplus, terms with small hierarchical surpluses can be excluded from the interpolation, or rejected. Some minimum value of $s_\mathbf{i}$ is chosen as the criterion for rejecting or retaining interpolation terms. Often a representation is constructed for the cross sections of several different nuclides, reaction types and energy groups, which may have different magnitudes. In order to use the same rejection criteria across all cross sections, the relative rejection criterium is defined as η such that

$$\frac{|s_\mathbf{i}|}{f(\mathbf{x_i})} < \eta, \tag{2.100}$$

where $\mathbf{i}$ is the multi-dimensional index of the node at sparse grid level q.

Chapter 3

Implementation

3.1 Special Considerations

The most popular method of implementing sparse grid quadrature or interpolation is to use the combinatorial formula, as formulated in Eq. (2.75) and Eq. (2.78). In this work, however, it is important to perform the interpolation and the approximation at both the highest level and the second-highest level for the internal error estimates that will be used. Using the combinatorial formula would therefore result in the function being sampled twice at some points. Given the computational cost of evaluating a cross section sample, it was considered imperative to find an algorithm that sampled each point only once, even if it is used repeatedly. Several papers mention in passing that this is possible when using a hierarchical construction, but no clear algorithm could be found that allowed one to calculate the quadrature weights or the hierarchical surpluses without sampling points from sparse grid levels $p \leq q$ repeatedly. In order to avoid this, an algorithm based on the difference formulas given in Eq. (2.37) was developed and implemented for sparse grid quadrature, as a part of the work done for this dissertation. An algorithm based on the van der Corput sequence was implemented for sparse grid interpolation (Bokov, Botes, and Zimin, 2012). Both algorithms are based on the principle of identifying each point in the grid uniquely with some vector of integer indices.

3.1.1 Calculating the quadrature weights

In Section 2.1.5.1 the delta quadrature operator, $\Delta Q_l[f(x)] = Q_l[f(x)] - Q_{l-1}[f(x)]$ was introduced and Eq. (2.37) was given as the formula for calculating the hierarchical quadrature weights. The delta quadrature operator can be expanded to find a more compact equation for the one-dimensional

hierarchical weights,

$$\begin{aligned}\Delta Q_l[f(x)] &= \sum_{i=1}^{n_l} w_{i,l} f(x_{i,l}) - \sum_{i=1}^{n_{l-1}} w_{i,l-1} f(x_{i,l-1}) \\ &= \sum_{i=1}^{2^l+1} w_{i,l} f(x_{i,l}) - \sum_{i=1}^{2^{l-1}+1} w_{i,l-1} f(x_{i,l-1}) \\ &= \sum_{i=1}^{n_l=2^{l-1}+1} \left(w_{i,l} - w_{\frac{1}{2}(i+1),l-1} \right) f(x_{i,l}),\end{aligned} \tag{3.1}$$

where $w_{\frac{1}{2}(i+1),l-1} = 0$ for i even, i.e. terms with non-integer indices are set to 0. This can be extended to multiple dimensions by a tensor product construction,

$$\Delta Q_{\mathbf{l}}^d[f(\mathbf{x})] = \bigotimes_{k=1}^{d} \left(\sum_{i_k=1}^{n_{l_k}} (w_{i_k,l_k} - w_{\frac{1}{2}(i_k+1),l_k-1}) \right) f(\mathbf{x}_{\mathbf{i},\mathbf{l}}), \tag{3.2}$$

therefore, in any given delta grid of level $\mathbf{j}$, the weight associated with any given point $x_{\mathbf{i},\mathbf{j}}$ is

$$\tilde{w}_{\mathbf{i},\mathbf{j}} = \prod_{k=1}^{d} (w_{i_k,j_k} - w_{\frac{1}{2}(i+1),l-1}). \tag{3.3}$$

This allows us to write Eq. (3.2) as

$$\Delta Q_{\mathbf{l}}^d[f(\mathbf{x})] = \sum_{\mathbf{i}} \tilde{w}_{\mathbf{i},\mathbf{l}} f(\mathbf{x}_{\mathbf{i},\mathbf{l}}). \tag{3.4}$$

Some way is sought to use Eq. (3.4) to transform Eq. (2.73), which is a sum over sparse grid levels, into a sum over points, i.e. the weights that would allow one to calculate

$$\mathbb{Q}_q^d[f(\mathbf{x})] = \sum_{\mathbf{x}_{\mathbf{i},\mathbf{l}} \in \mathbb{H}_q^d} w_{\mathbf{i},\mathbf{l}} f(x_{\mathbf{i},\mathbf{l}}). \tag{3.5}$$

In the practical implementation, a space in memory is reserved, in which the sparse grid will be built up from a "virtual" construction of each of these multi-dimensional quadrature operators, sequentially. This means that the tensor product grid is constructed, including weights, for a given vector of levels $\mathbf{l}$. Each of the grid points is then compared to each point that is already in the sparse grid memory space, using their vector indices, $\mathbf{i}$ and $\mathbf{l}$. If a point in the tensor product grid is found to have the same coordinates as a point in the sparse grid memory space, then the weight of the point in the tensor product grid is added to the cumulative weight of the corresponding point in the sparse grid. If, however, it is found that the point under consideration from the tensor product grid does not correspond to any point that is already in the sparse grid memory space, then this point from the tensor product grid and its weight are placed in the memory as a new sparse grid

point. A new vector $\mathbf{l}$ is then produced and the process repeated until the set of admissible indices is exhausted, whether that set contains the admissible indices for an isotropic or an anisotropic sparse grid.

The major problem raised in the preceding paragraph concerns the comparison process. Coordinates are usually real numbers, and comparing real numbers have well known pitfalls. For this reason a way has to be found to label points with unique integer numbers for comparison. To this end, note that the highest one-dimensional level in the sparse grid is easy to determine once the admissible set of indices has been fixed. Let us then consider the one-dimensional rule that has the highest one-dimensional level in the sparse grid, $\Delta Q_l[f(x)]$, and number the points in it sequentially from $x_1 = 0$ to $x_n = 1$ with integer numbers starting at 1. This will form our reference labels. We now have to find some mapping that, when applied to a rule $\Delta Q_j[f(x)]$ where $j < l$, will map the (sequential) label of the points from the lower level to the points with the same coordinates in the higher level rule.

If we note that, with the exception of $l = 1$ and $l = 2$, every point that is new to level l has an even index, and that all points with even indexes are new to level l, we can write down a relation between the index of a point at level l and the index of the same point at level $l + 1$:

$$i_{l+1} = 2i_l - 1. \tag{3.6}$$

In general, for any levels l, j where $l > j$, the following holds:

$$i_l = 2^{l-j}(i_j - 1) + 1, \tag{3.7}$$

where i_j is the index of a point in level j and i_l is the index of that same point in level l. This mapping can be used to replace the comparison of points by their coordinates with a comparison of indices by integer arithmetic. Every point has a vector of the one-dimensional levels at which it first appeared in each dimension and a vector of its indices in each dimension. This information is used in combination with Eq. (3.7) to determine the vector that contains the indices that the point would have in the highest one-dimensional level in every dimension.

The sparse grid quadrature, as implemented for this dissertation, is therefore constructed by the following steps:

- determine maximum one-dimensional level in the sparse grid;
- count the number of ways in which the one-dimensional levels can be combined to obtain the number of level vectors;
- count the number of points in the sparse grid of level q;
- sort the level vectors from the smallest $||\mathbf{l}||_1$ to the largest $||\mathbf{l}||_1$; and

- for each level vector,
  - count the number of points in the tensor product grid;
  - assign an index vector to each point in the tensor product grid;
  - calculate coordinates of each point in $\mathcal{I}^d$;
  - for each point: calculate the weights of the one-dimensional delta quadratures dimension-by-dimension, using Eq. (3.1), and find the total weight of that node by multiplying the one-dimensional delta weights;
  - use the transformation in Eq. (3.7) to find the index vector each point would have at the highest grid level; and
  - compare the index vector of every point with the index vectors of those points already in the sparse grid memory structure and if a match is found add the weights, else append the point to the end of the sparse grid memory structure.

3.1.2 Calculating the hierarchical surpluses

The nested Chebyshev grids can be built by using the binary van der Corput sequence (Corput, 1935). Consider a sequence of integer numbers $\nu = \{1, 2, 3, \ldots\}$. The van der Corput sequence is constructed by reversing the base 2 representation of the sequence ν, i.e.:

$$1_{10},\, 2_{10},\, 3_{10},\, 4_{10},\, 5_{10},\, \ldots = 1_2,\, 10_2,\, 11_2,\, 100_2,\, 101_2,\, \ldots \rightarrow$$
$$0.1_2, 0.01_2, 0.11_2, 0.001_2, 0.101_2, \ldots = \frac{1}{2},\, \frac{1}{4},\, \frac{3}{4},\, \frac{1}{8},\, \frac{5}{8}, \ldots \quad (3.8)$$

Note that in Eq. (3.8) the subscript "10" indicates base 10 and the subscript "2" indicates base 2 representation of a corresponding integer number. To each ν there corresponds a Chebyshev node, on the interval $\mathcal{I} = [0, 1]$, with the abscissa:

$$x_\nu = \frac{1}{2}\left[1 - \cos\left(\pi t_\nu\right)\right]. \quad (3.9)$$

Here parameter t_ν is defined as follows:

$$t_\nu = \sum_{j=1}^{l_\nu - 1} b_j 2^{-j} \quad (\text{for } \nu \geq 2), \quad (3.10)$$

where $b_j \in \{0, 1\}$ is the bit value in the binary representation of the integer number ν. The level l_ν of a node x_ν is defined as the position of the last (counting from the rightmost position) non-zero bit in the binary representation of integer ν plus 1.

As an example, consider $\nu = 6$. Its binary representation is $6 = 110_2$ which leads to: $l_6 = 3 + 1 = 4$, $t_6 = 0.011_2 = 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} = 3/8$ and $x_6 = \frac{1}{2}[1 - \cos(3\pi/8)]$. It should be emphasised

that the abscissa of a node, its level and the corresponding basis function are uniquely identified by a single index ν (Bokov, Botes, and Zimin, 2012).

Eqs. (3.9) and (3.10) do not, however, account for nodes $x = 0$ and $x = 1$. Therefore we introduce an extended sequence of integers $\nu \in \tilde{\mathbb{Z}}$ (where $\tilde{\mathbb{Z}}$ is defined as $\{-1, 0, 1, 2, 3, \ldots\}$) which corresponds to a modified van der Corput sequence. The modified van der Corput sequence accounts for the missing nodes, but also redefines the index number that corresponds to $x_1^0 = 1/2$ to preserve the order in which nodes are added. The redefined sequence is shown below with the modified terms underlined:

$$\underline{-1_{10}, 0_{10}, 1_{10}},\ 2_{10},\ 3_{10},\ 4_{10},\ 5_{10},\ \ldots \rightarrow \underline{\frac{1}{2}},\ 0,\ 1,\ \frac{1}{4},\ \frac{3}{4},\ \frac{1}{8},\ \frac{5}{8},\ \ldots. \tag{3.11}$$

by convention, $l_{-1} = 1$ and $l_0 = l_1 = 2$.

The interpolation grid $\mathcal{H}_l$ can now be constructed by incrementing ν, starting from $\nu = -1$. The abscissa x_ν and level l_ν of the node are calculated using the binary operations defined above and treating the first three terms (i.e. $\nu = -1, 0, 1$) as exceptions. The process is continued until the first node with a level $l_\nu = l + 1$ is achieved. The set of interpolation nodes and the delta set can, therefore, be defined as

$$\mathcal{H}_l = \{x_\nu : l_\nu \leq l\} \quad \text{and} \quad \mathcal{D}_l = \{x_\nu : l_\nu = l\}\,, \tag{3.12}$$

respectively.

3.2 Implementation of Sparse Grid Quadrature

As a part of the work done for this dissertation, a sparse grid quadrature module was developed and implemented in Fortran 90. It uses the method described in Section 3.1.1 to track the sparse grid nodes and calculate the quadrature weights of a sparse grid quadrature rule based on the one-dimensional Clenshaw-Curtis quadrature rule. A similar method was used to implement a Newton-Cotes sparse grid quadrature rule, for which results will not be reported in this dissertation. The accuracy and reliability of the sparse grid quadrature module was then verified in a systematic manner, as will be described below.

3.2.1 Testing procedure

The sparse grid quadrature was first tested as a stand-alone module on a set of standard test problems with known analytical solutions to verify the accuracy of the test code before using it for cross section approximation.

Table 3.1: The Genz test functions

Function name	Definition
oscillatory	$f_1(\mathbf{x}) = \cos\left(2\pi u_1 + \sum_{k=1}^{d} a_k x_k\right)$
product peak	$f_2(\mathbf{x}) = \prod_{k=1}^{d} \left(a_k^{-2} + (x_k - u_k)^2\right)^{-1}$
corner peak	$f_3(\mathbf{x}) = \left(1 + \sum_{k=1}^{d} a_k x_k\right)^{-(d+1)}$
gaussian	$f_4(\mathbf{x}) = \exp\left(-\sum_{k=1}^{d} a_k^2 (x_k - u_k)^2\right)$
continuous	$f_5(\mathbf{x}) = \exp\left(-\sum_{k=1}^{d} a_k \lvert x_k - u_k\rvert\right)$
discontinuous	$f_6(\mathbf{x}) = \begin{cases} 0, & x_1 > u_1 \text{ or } x_2 > u_2 \\ \exp\left(\sum_{k=1}^{d} a_k x_k\right), & \text{otherwise} \end{cases}$

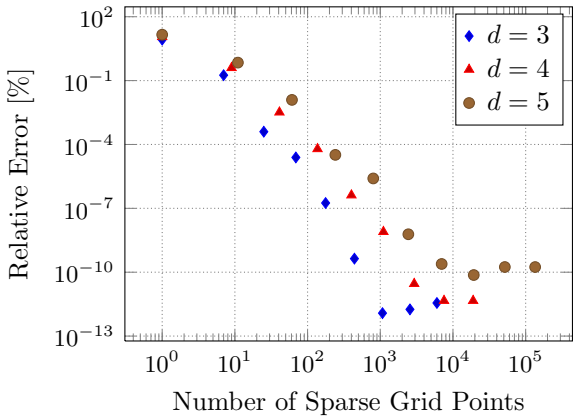
3.2.1.1 The Genz test functions

The sparse grid quadrature module was tested with several functions from the Genz test suite (Genz, 1984), for which analytic solutions are known. They will be named according to their behaviour, and these names will be used through the rest of the text. The Genz test functions are displayed in Table 3.1 and the analytic solutions of their integrals on the domain $[0, 1]^d$ in Table 3.2.

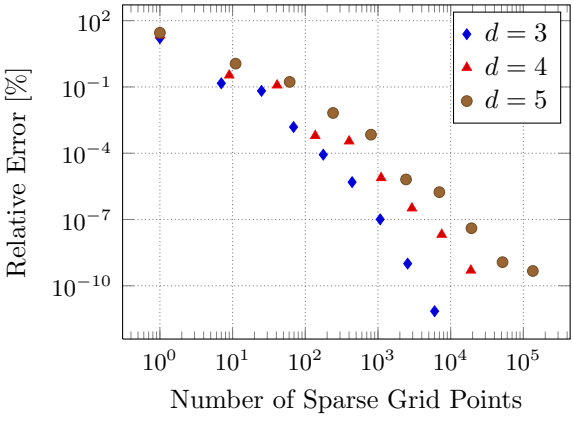
These functions were used to test the performance of both the isotropic and anisotropic versions of the sparse grid quadrature module. The corner peak function was not used as a part of the verification effort, since its analytic solution could not be coded correctly. The parameters a_k determine the difficulty that the function presents to quadrature methods, e.g. how fast it oscillates or how sharply it peaks, and must be chosen from $\mathbb{R}_+$. The parameters u_k are used to shift the function in space (Nahm, 2005) and are taken from $[0, 1]$. In this test the values were chosen arbitrarily as $a_k = 0.8$ for all $k = 1, \ldots, d$ and $u_k = 0.5$ for all $k = 1, \ldots, d$.

3.2.1.2 Error analysis

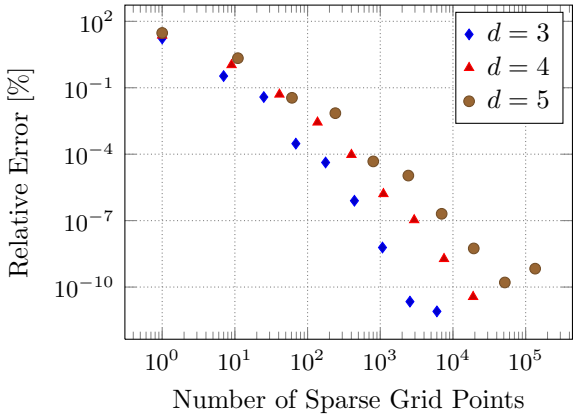
The sparse grid quadrature module was used to numerically integrate the Genz test functions in 3, 4 and 5 dimensions, using sparse grid levels 0 through 8. The analytic solutions to the integrals in Table 3.2 were then used to determine the error of the sparse grid quadrature. Results for the corresponding isotropic sparse grids are presented in Figure 3.1. Note the relatively fast convergence for the very smooth functions, e.g. Figure 3.1(a) as opposed to the slow convergence for the



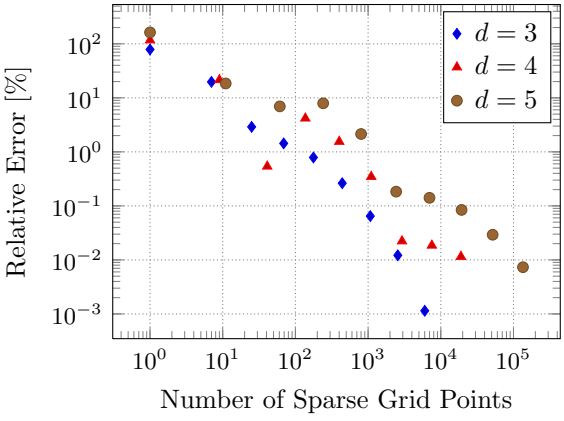
(a) Oscillatory



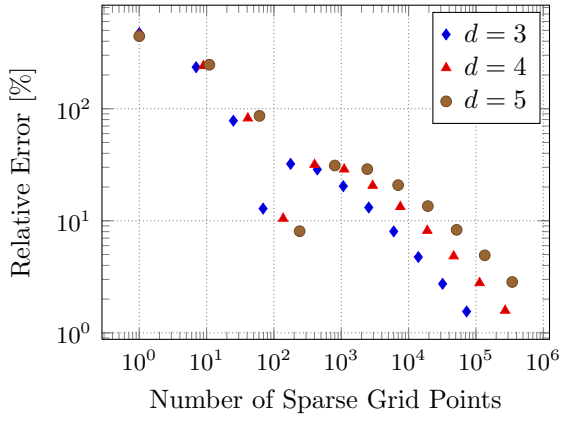
(b) Product Peak



(c) Gaussian



(d) Continuous



(e) Discontinuous

Figure 3.1: Relation between the error and the number of points used to perform quadrature on the Genz functions. In each case the level increases by 1 at each new data point

Table 3.2: The analytical solutions to the definite integrals of the Genz test functions on the interval $[0,1]^d$

Function name	Solution
oscillatory	$I[f_1(\mathbf{x})] = 2^d \cos\left[\frac{1}{2}\left(4\pi u_1 + \sum_{k=1}^{d} a_k\right)\prod_{k=1}^{d}\sin\left(\frac{a_k}{2}\right)\right]\left(\prod_{k=1}^{d} a_k\right)^{-1}$
product peak	$I[f_2(\mathbf{x})] = \prod_{k=1}^{d} a_k\left[\arctan(a_k(1-u_k)) + \arctan(a_k u_k)\right]$
corner peak	$I[f_3(\mathbf{x})] = \left(d!\prod_{k=1}^{d} a_k\right)\sum_{\alpha\in\{0,1\}^d}\frac{(-1)^{\|\|\alpha\|\|_1}}{1+\sum_{k=1}^{d}\alpha_k a_k}$
gaussian	$I[f_4(\mathbf{x})] = \prod_{k=1}^{d}\frac{\sqrt{\pi}}{2a_k}\left[\mathrm{erf}\left(a_k u_k\right) - \mathrm{erf}(a_k(u_k-1))\right]$
continuous	$I[f_5(\mathbf{x})] = \prod_{k=1}^{d} a_k^{-1}\left[2 - \exp\left(-a_k u_k\right) - \exp\left(-a_k(1-u_k)\right)\right]$
discontinuous	$I[f_6(\mathbf{x})] = \left(\prod_{k=1}^{\min(2,d)}\frac{\exp\left(a_k u_k\right)-1}{a_k}\right)\left(\prod_{k=3}^{d}\frac{\exp\left(a_k\right)-1}{a_k}\right)$

Table 3.3: Number of sparse grid points in an isotropic grid with 3, 4 and 5 dimensions

d \ q	0	1	2	3	4	5	6	7	8
3	1	7	25	69	177	441	1073	2561	6017
4	1	9	41	137	401	1105	2929	7537	18945
5	1	11	61	241	801	2433	6993	19319	51713

discontinuous function, Figure 3.1(e), which is typical behaviour for sparse grid quadrature. The number of samples in an isotropic sparse grid at each level is given in Table 3.3 for all three test cases.

A similar analysis was performed for anisotropic sparse grids using a variety of anisotropy vectors. This analysis is not shown here, since the results were not substantially different from the results for the isotropic case. This is because, except for the discontinuous function, the Genz test functions are inherently isotropic functions.

3.3 Implementation into the Cross Section Representation Code

After the accuracy and reliability of the sparse grid quadrature module were established in Section 3.2.1, it was coupled to a pre-existing code that performed cross section representation through approximation. This code receives the generated sparse grid points from the sparse grid quadrature

module and samples the few group neutron cross sections at these points by calling a transport solver. The code then uses the sparse grid quadrature module to perform quasi-regression and thus create a polynomial cross section library. Before the new sparse grid quadrature module was coupled to that representation code, a module called SANDIA SPARSE, produced by John Burkardt of the Florida State University (Burkardt, 2009) was used. The results from cross section approximation using the new sparse grid quadrature module compared well to results that were obtained previously with SANDIA SPARSE.

It was considered necessary to produce an in-house sparse grid quadrature code for several reasons, the most important of which is licensing considerations. SANDIA SPARSE is distributed under the GPL licence, which is quite suitable for use in research as has been done up to now, but the cross section representation methodology will eventually be implemented into and distributed as part of a proprietary reactor analysis code system developed at Necsa. A secondary consideration was to have a module that could be changed and adapted as necessary. It should also be noted that our implementation of the Newton-Cotes sparse grid quadrature was more numerically stable than the implementation of the same in SANDIA SPARSE.

To perform interpolation, the cross section approximation code was used to obtain samples at the necessary sparse grid points from the assembly code. These samples were then written to a file, which was read by a separate program that performed the interpolation.

3.3.1 Error estimation

3.3.1.1 Built-in measures

The relation between the variance of a function and the truncation error in its approximation was discussed in Section 2.4. In the same section, it was shown that the magnitude of the hierarchical surpluses may be used to estimate the maximum error of interpolation. The results from such analysis are very important from the perspective of model optimisation and may provide important insights into the model and the behaviour of the function that is being represented. One does, however, also wish for the representation method to deliver a error estimate that is simple and easy to interpret, constructed from the information available from the samples that have already been used to construct the representation.

Different cross sections may vary in their magnitude and therefore in the absolute error measures from Section 2.4. It is therefore preferable to use a relative error estimate. Cross sections may also be zero in some samples, or in a small region in the parameter space, e.g. cross sections of fissionable isotopes with a threshold energy. This situation needs to be dealt with carefully. To avoid division by zero errors, the relative error estimates are normalised to the function average, which will always be non-zero unless the cross section is zero on the whole parameter space, instead of the function value at any sample point.

In this section, we use the definition of the average or expectation value, Eq. (2.89), and introduce the Average Relative Error (ARE) and the Maximum Relative Error (MRE):

$$\mathrm{ARE} = \frac{E\left|f(\mathbf{x}) - f_N(\mathbf{x})\right|}{E\left[f(\mathbf{x})\right]}, \tag{3.13}$$

and

$$\mathrm{MRE} = \max_{\mathbf{x}\in\mathcal{I}^d} \frac{\left|f(\mathbf{x}) - f_N(\mathbf{x})\right|}{E\left[f(\mathbf{x})\right]} \tag{3.14}$$

which includes a normalisation by the average of the approximated function. The expectation value can be calculated by performing a sparse grid quadrature using the function samples that have already been used to construct the representation.

Few group neutron cross sections can be zero across the whole parameter space, e.g. the fission cross section of a non-fissionable isotope. One should therefore be careful of dividing by the function value, or the function average. This case is easy to deal with, since the variance, as calculated by solving the integral in Eq. (2.92) numerically, will be exactly zero. If a cross section has zero variance and any sample of that cross section is non-zero, it is assumed the function is constant. Therefore error measures are only calculated for a cross section if it has non-zero variance or if any sample from the cross section is not zero.

3.3.2 Independent error estimation

A part of this work was to validate the built-in error measures described in the previous section. In order to verify the cross section representations that were produced, it was necessary to compare the representation with the true function value at a large number of uniformly distributed points, that were not used to produce the representation. This comparison included the calculation of several error measures and will be referred to from here onwards as *post-representation processing*, which is not a part of the representation method and would not be used under normal circumstances. It was implemented for this work to verify various aspects of the representation method, especially the built-in error measures.

The post representation processing code uses a Sobol' sequence (Sobol', 1993) to generate the uniformly distributed test points. The Sobol' quasi-random number generator by John Burkardt (Burkardt, 2009) was used to generate the quasi-random points that were used for testing. This module was interfaced with the cross section representation code in such a way that it could use the transport solver to calculate cross sections at the quasi-random points, without proceeding to the representation step. The transport solver produces a file that contains, amongst other things, the point-wise cross section values. A routine that reads this file and then uses the cross section values at the quasi-random test points to calculate the necessary error estimates was implemented. The

expectation value is estimated by

$$E[f(\mathbf{x})] = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i) \tag{3.15}$$

where N is the number of test samples. The average relative error as measured on the independent samples is calculated as

$$\text{ARE} = \frac{1}{N}\sum_{i=1}^{N} \frac{|f(\mathbf{x}_i) - f_N(\mathbf{x}_i)|}{E[f(\mathbf{x})]}, \tag{3.16}$$

and similarly the maximum relative error is calculated as

$$\text{MRE} = \max_{1\leq i\leq N} \frac{|f(\mathbf{x}_i) - f_N(\mathbf{x}_i)|}{E[f(\mathbf{x})]}. \tag{3.17}$$

3.4 Work not Reported Elsewhere

3.4.1 Sparse grid based on equidistant nodes

Interpolation and approximation on Chebyshev nodes was described in Chapter 2. Before turning to Chebyshev nodes, however, the sparse grid quadrature module implemented for this dissertation was based on Newton-Cotes quadrature, which uses an equidistant one-dimensional grid. The sparse grid quadrature that was based on the Newton-Cotes rule produced acceptable results when applied to the Genz test functions above. However, when it was used as the quadrature module in the cross section representation code, results that were obtained for this quadrature were unexpectedly poor. In the hope of improving the performance of the Newton-Cotes sparse grid quadrature, Richardson extrapolation was implemented as an acceleration technique (Burden and Faires, 2001). The accelerated Newton-Cotes sparse grid gave results for the Genz functions that were an improvement on those obtained without acceleration, but when applied to cross section approximation the results were again disappointing. At this point in the development of the sparse grid quadrature module, Newton-Cotes quadrature was replaced by the Clenshaw-Curtis based method discussed in this work.

The equidistant one-dimensional grid that is used by Newton-Cotes quadrature was also used for sparse grid interpolation using piecewise linear basis functions. It was applied to a pre-parametrised polynomial test problem and the results were reported in (Botes and Bokov, 2011). The results were promising, but a piecewise linear representation comes with an important drawback: it suffers from unacceptable discontinuities in the first derivatives of the interpolation functions. These discontinuities are a concern, given the necessity of calculating quantities such as reactivity coefficients (Bokov, Botes, and Zimin, 2012). It was decided not to include piecewise linear sparse grid interpolation in this work in order to limit the scope, since it was an intermediary step along the path to implementing Lagrange interpolation on a sparse grid based on Chebyshev nodes.

Some development on the part of the cross section representation code was necessary in order to prepare it for this study. Various pieces of functionality were added or amended, most of which are not relevant to the results that will be presented, but that were critical to make the code functional after it had fallen out of use for some time. One of the most important developments was to re-establish the interface with the existing assembly code, so that cross sections could be sampled at the points dictated by the sparse grid generator and returned for further processing. Additionally, the cross section representation code in its original form could only represent capture cross sections. Most of the results that will be presented are for capture cross sections, but before the study was undertaken the ability to represent fission, nu-fission and transport cross sections were implemented. As a contribution to other, unrelated work, representation of macroscopic scattering cross sections for an arbitrary number of energy groups was added to the functionality of the representation code. Data transfer via files from the main body of the cross section representation code, which is mainly an approximation by quasi-regression code, to the free-standing interpolation code was also implemented.

Chapter 4

Results

4.1 Problem Description

4.1.1 The example problems that were investigated

To test the performance of the approximation and interpolation processes in representing cross sections, three test problems were considered: a PWR fuel assembly, an MTR fuel assembly and a VVER fuel pin. These examples will be described in more detail later, but some basic information is contained in Table 4.1. Though all three examples are light water reactors, MTR fuel is significantly different from the VVER and PWR fuel, for several reasons:

- the operating conditions differ from those of the power reactors, which would change the way the cross sections depend on the state parameters;
- the geometry is quite different, since the fuel is divided into plates and not pins; and
- most importantly the fuel is 90 % enriched and it is burned to a much higher burnup before leaving the core, therefore the dependence on burnup will be distinct from the low enriched power reactor fuel and a larger number of fission products need to be accounted for.

Table 4.1: Details of the assemblies that were investigated

Type	PWR fuel assembly	MTR fuel assembly	VVER fuel pin
Enrichment	3.25 %	90 %	4.4 %
Geometry	17×17 pins	19 plates	1 pin
Model symmetry	$1/8$	$1/4$	cylindrical
Boron	yes	no	yes
Microscopic isotopes	16	37	2

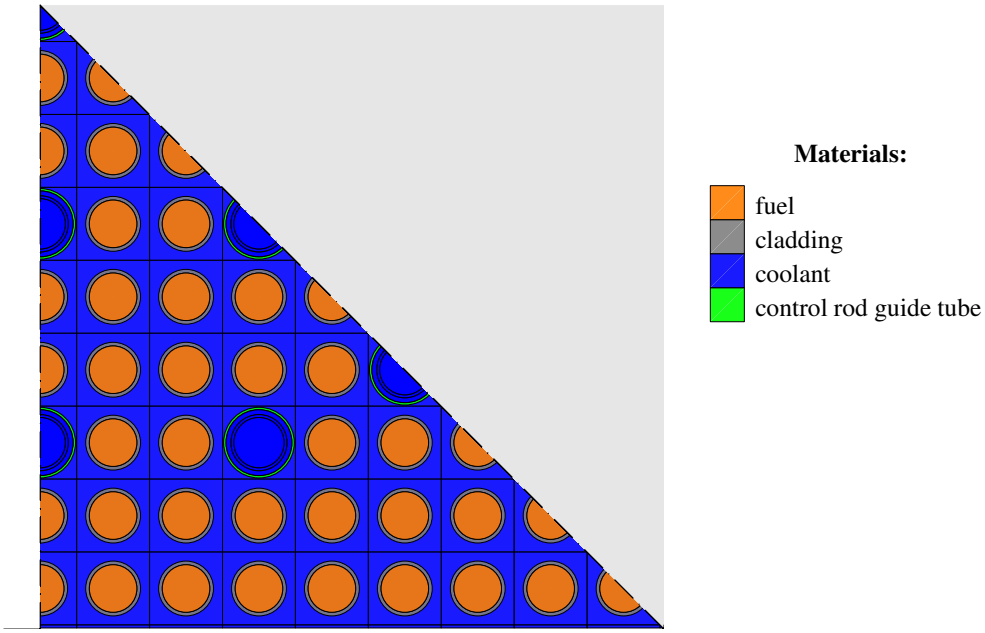


Figure 4.1: One-eighth symmetry segment of a PWR fuel assembly

These three diverse problems were chosen to show the robustness of the method.

The PWR example is a standard 3.25 % enriched fuel assembly with all control rods extracted and no burnable absorbers. It was modelled in one-eighth symmetry, and a diagram of the geometry as it was modelled in the transport code is shown in Figure 4.1. It measures 21.6 cm on each side and contains fuel pins in a 17×17 lattice. The cross sections were collapsed to six energy groups, with the structure shown in Figure 4.4. The fuel is UO_2, the cladding is Zircalloy-4 and the coolant is borated light water. The dashed lines represent symmetry axes and the area that is shaded grey indicates the part of the assembly that would have to be added to obtain a quarter symmetry model. The state parameters that were employed for this example are burnup, fuel temperature, moderator temperature, moderator density and soluble boron concentration.

The MTR example is a standard MTR HEU (highly enriched uranium) fuel assembly, enriched to 90 %, as was used in the SAFARI-1 research reactor before its conversion to low-enriched fuel. It was modelled in one-quarter symmetry and a diagram of the geometry as it was modelled in the transport code is shown in Figure 4.2. The model used the same six-group energy structure as the PWR example, as shown in Figure 4.4. The assembly is 8.1 cm by 7.71 cm and contains 19 parallel fuel plates that consist of Aluminium cladding and a 90 % enriched Uranium-Aluminium alloy as fuel. The coolant is light water and the dashed lines again indicate symmetry axes. The state parameters that were employed for this example are burnup, fuel temperature, moderator temperature and moderator density.

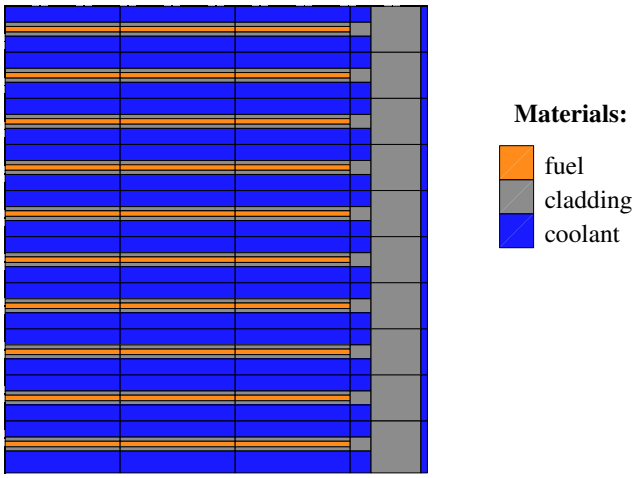


Figure 4.2: One-quarter symmetry segment of an MTR fuel assembly

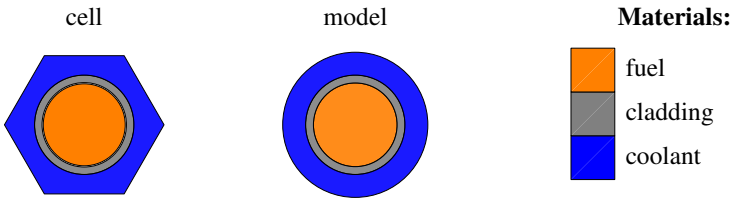


Figure 4.3: Representation of a VVER fuel pin (left) and the cylindrical model (right) used in the assembly code

The VVER example is a single 4.4 % enriched fuel pin with natural Zirconium cladding, UO_2 fuel and borated light water as coolant. A diagram of the geometry as it was modelled in the transport code is shown in Figure 4.3 and further details can be found in Zimin and Semenov (2005). This example utilised a two-group energy structure, as shown in Figure 4.4. The state parameters that were employed for this example are burnup, fuel temperature, moderator temperature, moderator density and soluble boron concentration, the same as for the PWR assembly.

Results are presented for both the MTR and PWR example's group six macroscopic absorption cross section throughout, and for the VVER pin's thermal macroscopic absorption cross section. Though microscopic cross sections are also available, results for macroscopic cross sections are presented as they display a more complex dependence on the state parameters (Müller, 1987), as discussed in Chapter 1. The dependence of thermal group cross sections on the state parameters are also known to be more complex than that of fast group cross sections (Zimin and Semenov, 2005). The results presented are therefore for the most challenging examples. The study, as described here, was performed for various other cross sections and important isotopes and some of these results will be discussed where they may be of value.

The parametric study of the correlation between accuracy and the number of samples goes to extremes to demonstrate the potential of the method. It shows that the user can control the accuracy in any application by increasing the number of samples. To evaluate the method for use in

a specific application some minimum accuracy must be specified. This study defines "acceptable" error for day-to-day reactor support calculations to be MRE =0.1 % – since this is the maximum error accepted in the current method, as estimated on the samples used for the fit – and ARE =0.01 %, from consultation with experts. It should be pointed out that a maximum relative error of less than 0.1 % as measured by the built-in error estimate is a far more stringent requirement than the same limit on MRE in the current cross section representation method used at Necsa.

These limits on the error are very stringent, and in fact several experts expressed the opinion that limits of MRE =0.2 % and ARE =0.05 % are quite acceptable for reactor calculations as they are performed at present. The objective of this project is, however, not to replicate what is acceptable at present, but to explore the way forward for cross section representation as it may be required in future, therefore stringent limits, that are expected to become the norm in future, were used. It is always necessary to balance the demands of higher accuracy against factors that make accuracy harder to achieve, such as the state parameter intervals on which the representation is valid and the number of dimensions that should be included. The use of sparse grids may mitigate some of these factors, but will not eliminate them.

It should also be noted that the PWR example was designed to be challenging to any representation method, with six groups and large state parameter intervals. Furthermore, the results of the most complex cross sections will be presented in this dissertation and these results could be viewed as a "limiting case" for the representation of cross section dependencies in a standard PWR assembly. Adding extra complications, such as burnable poisons, is planned as future work.

4.1.2 Details of the transport calculations

All the transport calculations were performed in two dimensions. The PWR and MTR examples were calculated with the HEADE (HEterogeneous Assembly DEpletion) collision probability code (Reitsma and Joubert, 1999) and the VVER pin was calculated with the UNK method of characteristics code (Davidenko and Tsibulsky, 1998).

Power reactors traditionally use only two energy groups, but bearing in mind the impact of newer fuel designs, this simple energy treatment will eventually become insufficient. There is already evidence that two energy groups are insufficient to treat the spectrum effects of MOX (Downar, Lee, and Jiang, 2000). Since this work was done specifically to keep up with more sophisticated calculational models that will be required in future, both microscopic and macroscopic cross sections were produced and a more challenging group structure with six energy groups was used for the PWR and MTR examples. The group structure is set out in Figure 4.4. Groups 5 and 6 are the thermal groups, groups 3 and 4 the epithermal groups and groups 1 and 2 the fast groups.

Depletion calculations were done at a set of chosen nominal conditions, then off-bases were done at most of the burnup steps to obtain the necessary samples that deviate from nominal conditions

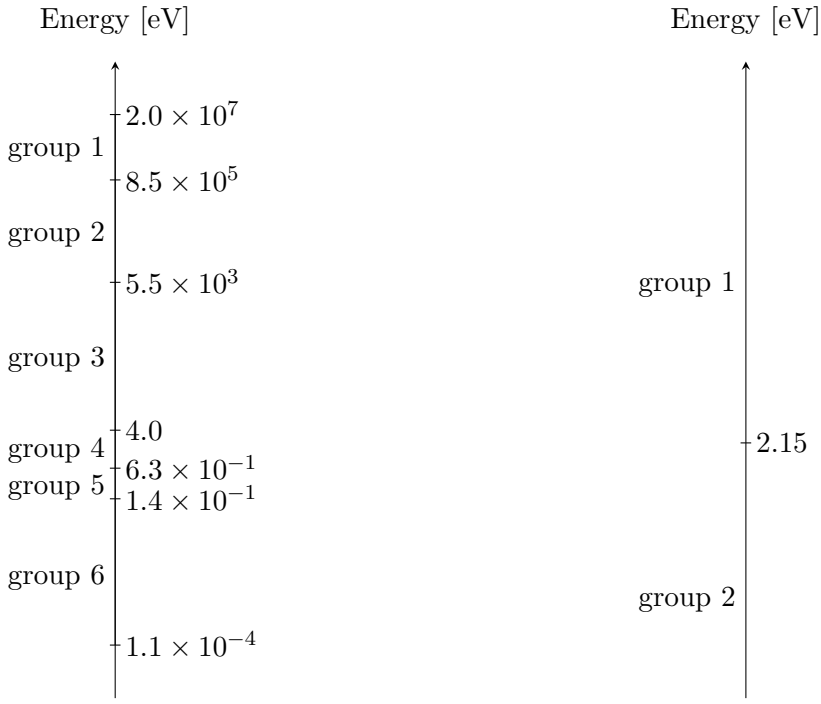


Figure 4.4: Energy group structure used in the calculations. The six-group structure used for the PWR and MTR examples is on the left and the two-group structure used for the VVER example on the right

Table 4.2: Nominal conditions and intervals of variation for the PWR assembly

	nominal	min	max
Burnup [MWd/tU]	N/A	0.0	60000
Moderator density [g/cm^3]	0.67	0.27	0.97
Moderator temperature [K]	579.4	279.4	979.4
Fuel temperature [K]	951.4	291.4	1651.4
Boron concentration [ppm]	600	1	1600

in the other dimensions. Table 4.2 gives these nominal conditions as well as the state parameters' interval of variation for the PWR case. Table 4.3 provides the same information for the MTR case and Table 4.4 for the VVER case.

The intervals of variation of the parameters in the MTR case are chosen such that they cover the state of the core from cold start-up to normal operational conditions and mild transients. The library that results from a calculation with these limits can then be viewed as an operational library. The objective is, however, to have a method that can create libraries suitable to any situation. To test the robustness of the method, the intervals of variation for the PWR library were therefore chosen to cover core conditions at a wider variety of possible states, to be applicable to possible transient calculations.

Table 4.3: Nominal conditions and intervals of variation for the MTR assembly

	nominal	min	max
Burnup [MWd/tU]	N/A	0.0	450000
Moderator density [g/cm^3]	0.99	0.59	1.09
Moderator temperature [K]	318.0	278.0	448.0
Fuel temperature [K]	333.0	283.0	533.0

Table 4.4: Nominal conditions and intervals of variation for the VVER fuel pin

	nominal	min	max
Burnup [MWd/tU]	N/A	0.0	60000
Moderator density [g/cm^3]	0.72	0.4	1.05
Moderator temperature [K]	575	293	603
Fuel temperature [K]	950	293	1793
Boron concentration [ppm]	500	0	3000

Not all burnup steps have off-bases, since it is possible to specify a "minimal" burnup mesh independently of the samples that are required for the sparse grid. The extra burnup points were not used for the representation, but selecting them based on sound engineering considerations ensured that error in the depletion calculation did not contaminate the results regarding the accuracy of the representation.

4.1.3 Cross section models

An assembly code may produce macroscopic cross sections, containing the effect of all nuclides from the fine group library, and microscopic cross sections for individual nuclides, which are sometimes called microscopic materials. Which cross sections are produced often depends on the specific problem being considered. In this study, the model of the MTR assembly uses 37 microscopic materials, while 16 are used for the PWR assembly model and 2 for the VVER pin model. The larger number of microscopic materials in the MTR model is to account for the fission products that result from the high burnup level reached by HEU MTR fuel before it is disposed of.

The materials that were investigated are set out in Figure 4.5 for the PWR example and in Figure 4.6 for the MTR example. These figures show three macroscopic materials for each reactor model: "Water", "Structural" and "All". The material marked "All" is the macroscopic material which contains the effects of all the materials that are present in the fine group library. The material marked "Structural" accounts for all the materials that are not treated microscopically, with the exception of water, which is represented on its own in each case. It should also be pointed out that the VVER cross sections contained, among the standard macroscopic cross sections, two microscopic cross sections: thermal absorption of $^{135}_{54}$Xe and thermal absorption of $^{149}_{62}$Sm.

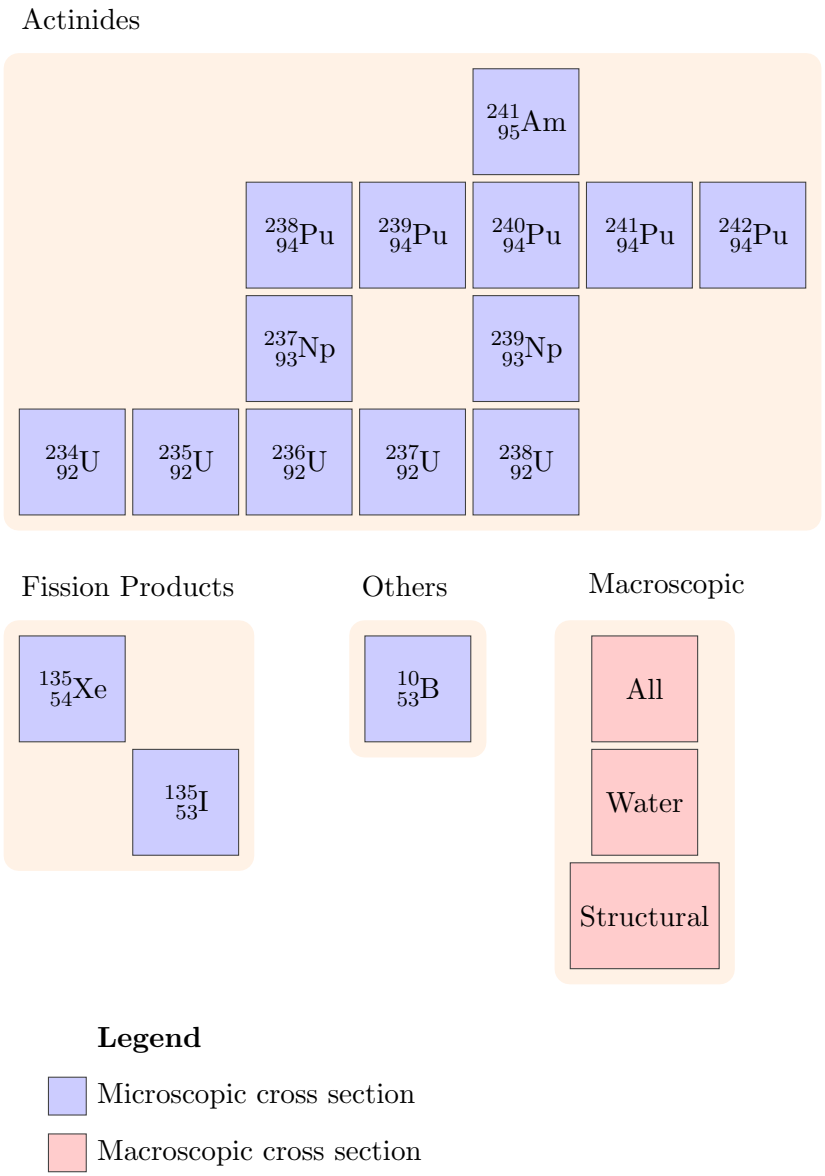


Figure 4.5: Microscopic and macroscopic materials used in the PWR example

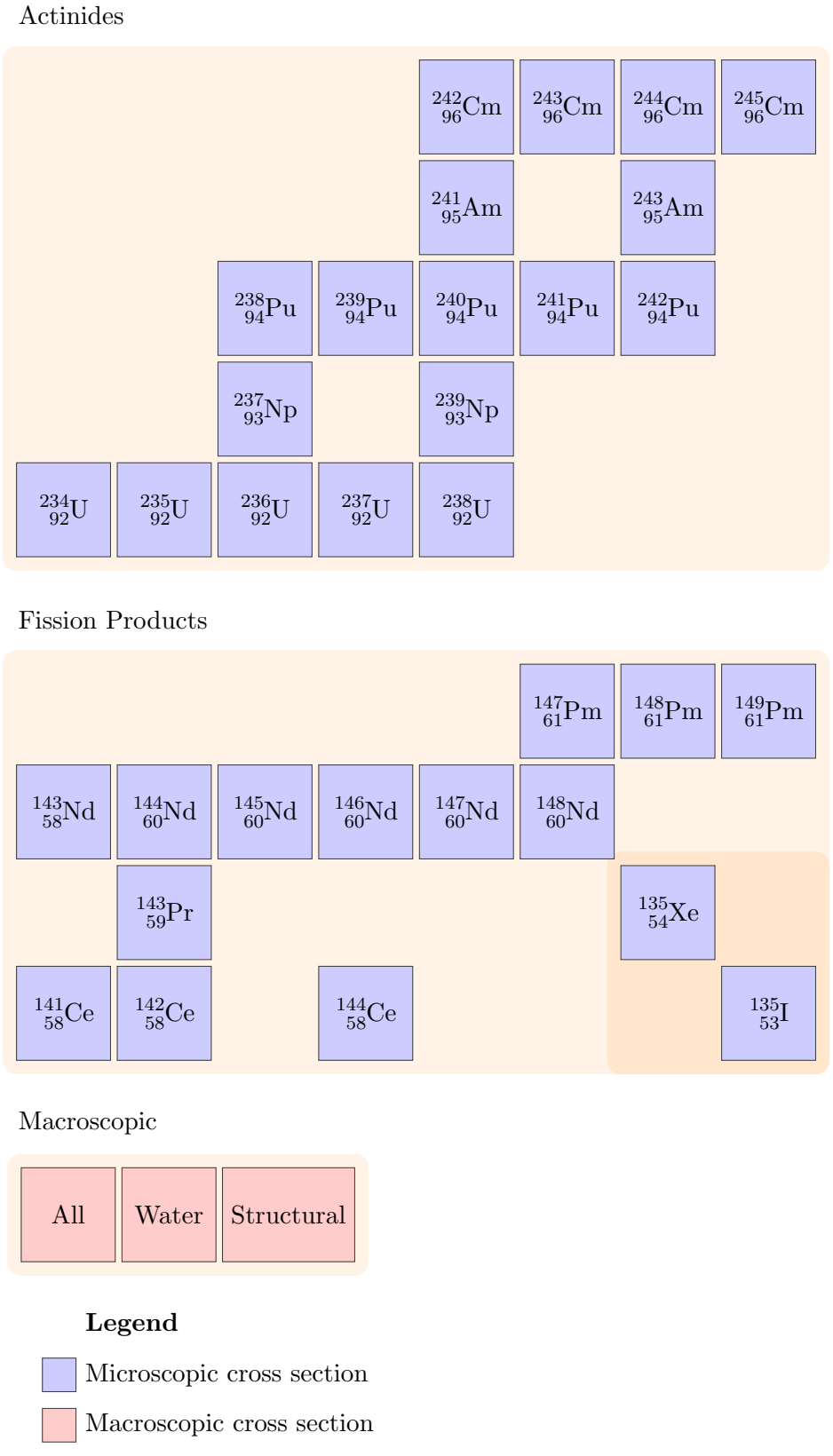


Figure 4.6: Microscopic and macroscopic materials used in the MTR example

4.2 Applying the Method to the Problem

4.2.1 Representation limits and error estimation

Approximation of the few group neutron cross sections was done by performing quasi-regression, as described in Section 2.1.1.2, using Legendre polynomials as basis functions. The regression coefficients were obtained through sparse grid quadrature, using the point-wise cross section values produced by the transport code.

The representations were shrunk to optimise them in terms of storage requirements and reconstruction speed. This step of shrinking the representation is absolutely necessary in order to produce a cross section *library* (the file that contains the information necessary to reconstruct a cross section) that is both practical and accurate. In the case of interpolation, there are as many terms in the representation as there are nodes in the sparse grid that was used to construct the interpolation. Sparse grids of high level may contain several thousand nodes. Approximation has a different set of limits on the number of terms in the polynomial expansion, as explained in Section 2.2.4. These limits were set to $B_0 = 3$, $B_1 = 25$ and $B_\infty = 25$. In the five-dimensional cases (PWR and VVER) this choice of limits on the set of basis function indices results in 13400 possible combinations of the various i_k, in other words 13400 terms in the expansion. In the four-dimensional case (MTR) this would result in 11100 terms. Such a number of expansion terms would require very large amounts of memory to process. It is therefore clear that an optimisation step is essential for both approximation and interpolation.

To ensure that all complexities in the model could be captured, a high order of 25 was used for the basis polynomials in the approximation. However, it will be demonstrated later that even this high order was not completely sufficient in some cases. The optimisation was performed by rejecting terms based on the criteria described in Section 2.4.1.

Lagrange interpolation as discussed in Chapter 2 was also done for the example problems. The necessary step of shrinking the representation was performed according to the criteria described in Section 2.4.2, with $\eta = 1 \times 10^{-5}$.

Error was estimated by sampling a large number of independent, uniformly distributed test points, using the methods discussed in Section 2.4. The test points were based on a Sobol' sequence (Sobol', 1967), which ensured a uniform distribution and good coverage of the parameter space. These samples were used to calculate the error measures described in Section 3.3.1 during post-representation processing.

4.2.2 Sampling strategy

In all the cases and examples that will follow, sparse grids of a relatively high level were used. Due to the nestedness property of the sparse grids employed in this study, a sparse grid of level q contains all the nodes of any sparse grid with level $p \leq q$. Exploiting this property avoided the need to re-run the transport code every time a new sparse grid level was used.

There are several reasons why a wide range of sparse grid levels was chosen and such a large number of samples calculated, more than will be necessary or feasible to use in everyday calculations. On the one hand, the aim of using this range of sparse grid levels is to demonstrate the efficiency of the sparse grid sampling scheme and thereby indicate that accuracy that is acceptable for use in day-to-day reactor support calculations may be obtained by using a relatively small number of samples. On the other hand, it is also interesting to demonstrate the flexibility that is available for application to problems that are particularly complex or require a high degree of accuracy.

The significant body of experience of operating light water reactors allows one to tailor a method by taking into account existing knowledge, such as the complexity of cross section dependencies on burnup. Some examples of such tailoring, including splitting the burnup interval and sampling preferentially from burnup, will be demonstrated further on in the results. Combining efficient representation methods with the tailoring made possible by such knowledge may lead to highly accurate representations. What is considered to be acceptable accuracy is, to a certain extent, defined by what is feasible. Once more accurate representations are readily available, they will become the new standard for what is acceptable error.

Furthermore, the intention of using the variety of examples and sparse grid constructions included in this work is to demonstrate the potential that is available for application to non-standard problems. Other reactor designs may not have the benefit of experience that Light Water Reactors have and in such a case one would like to apply the few group cross section representation methodology as a “black box” and use it to extract certain information. In order to identify relevant state parameters, it is possible to construct a representation with a large number of parameters as input and then to use ANOVA to analyse the relative importance of each term, as well as the number of interactions between parameters. From this representation, one could then extract a “minimal” set of state parameters with which to create cross section representations for future applications. One could also identify the parameters that require special consideration, and may be used to refine the representation for application in the design process and, eventually, to support reactor operations. As such, in the absence of a transport code that could use more state parameters or model more exotic reactor designs, calculations up to a high sparse grid level were performed to demonstrate that it is possible to use sparse grid methods on large problems.

4.3 The Standard Case

In the examples that will be used as a basis, and referred to throughout the document as the standard case, the buildup of Xenon at the beginning of the assembly's life was followed explicitly. Treating Xenon in this way is artificial, since the assembly will experience Xenon fluctuations repeatedly during its lifetime. Xenon transients occur at the start-up of every cycle, and during possible emergency shutdown situations. A fuel element is usually placed in the core for several cycles.

Only the PWR and MTR examples will be used in the standard case, the VVER example will be introduced in one of the variations in the calculational approach that will follow.

4.3.1 Calculating the sparse grid samples

The sample points that correspond to a level eight isotropic sparse grid were generated and passed to the transport code, for both the PWR and MTR examples. In the PWR example, which has five dimensions, level eight corresponds to 51713 samples and in the MTR example, which has four dimensions, level eight corresponds to 18945 samples. The number of nodes contained in lower sparse grid levels was given in Table 3.3 in the previous chapter. The points and function values were then used to construct either an approximation or an interpolation of the cross section dependencies at each sparse grid level.

Level eight was found to be the upper limit of what is feasible for the PWR problem, as a result of the limits of the transport code that was used and the available computing equipment. All of the calculations, used for the MTR and PWR examples in this study, were done on a computer that has an Intel quad core processor with each core clocked to 2.66 GHz, 3.8 GB of physical memory and 2.0 GB of swap space running Ubuntu 11.10. Level eight was the highest level that could be calculated using this hardware configuration without the transport code running out of memory.

4.3.2 Accuracy of the representation methods in the PWR example

The accuracy of the approximation for the standard PWR problem can be seen in Figure 4.7, and that of the equivalent interpolation can be seen in Figure 4.8. The two cases are considered to be directly comparable since the same set of sparse grids were used for both, that is isotropic sparse grids of levels q from three to eight. There are several notable aspects to these figures, the first being that interpolation offers consistently better accuracy than approximation, as measured by both average relative error (ARE) and maximum relative error (MRE). It is also clear from these figures that the inherent error measures of both methods closely follow the various errors as measured on the independent samples. Unfortunately, the required accuracy of 0.01 % for average relative error is not reached, even at the highest level, for either interpolation or approximation. The target of

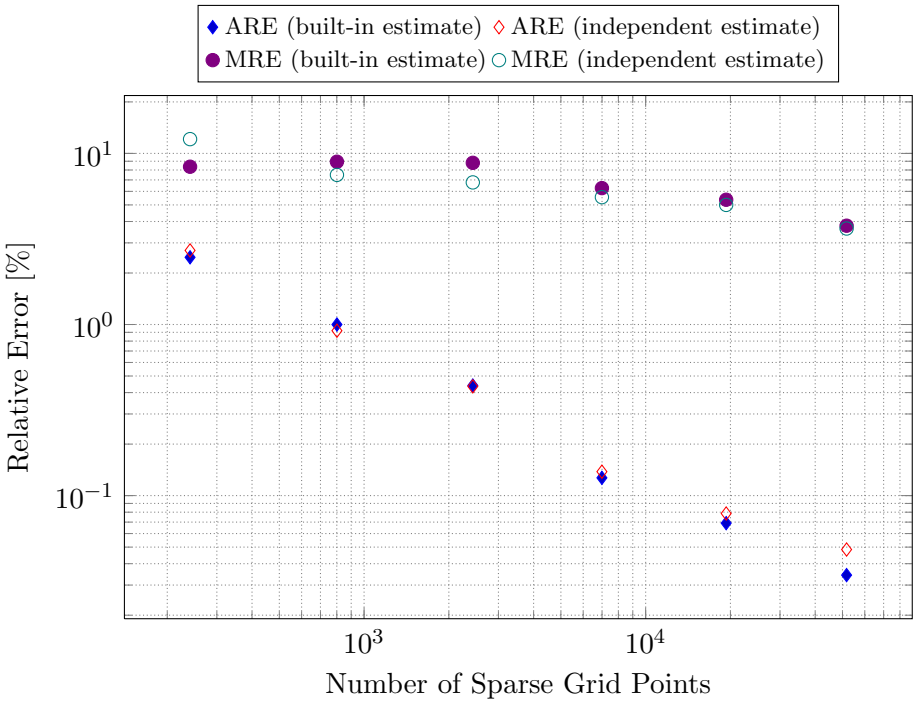


Figure 4.7: PWR assembly: approximation errors in the standard case

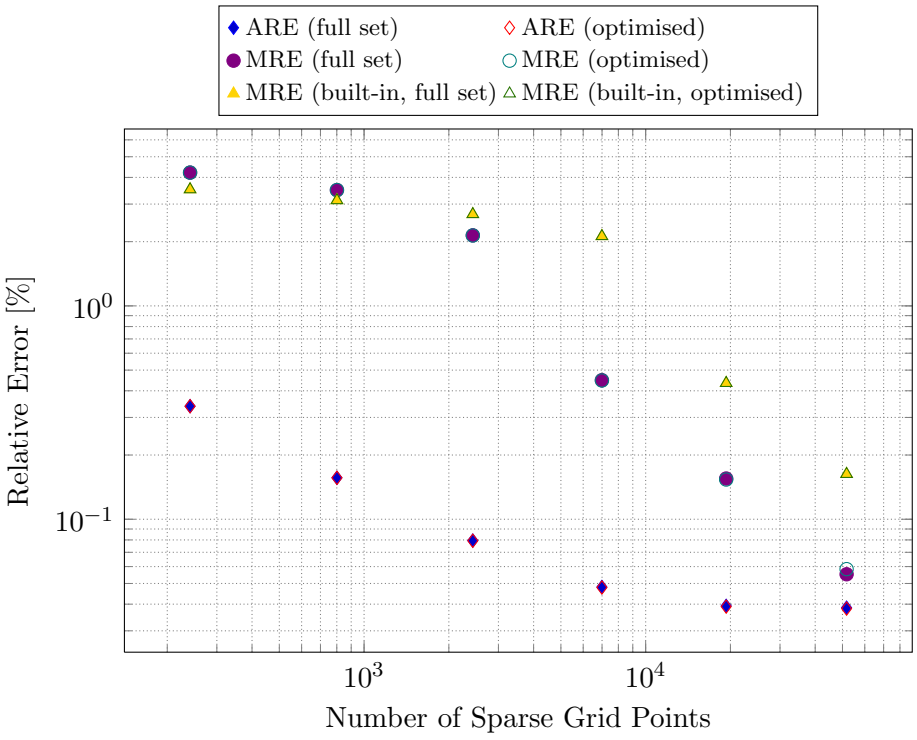


Figure 4.8: PWR assembly: interpolation errors in the standard case

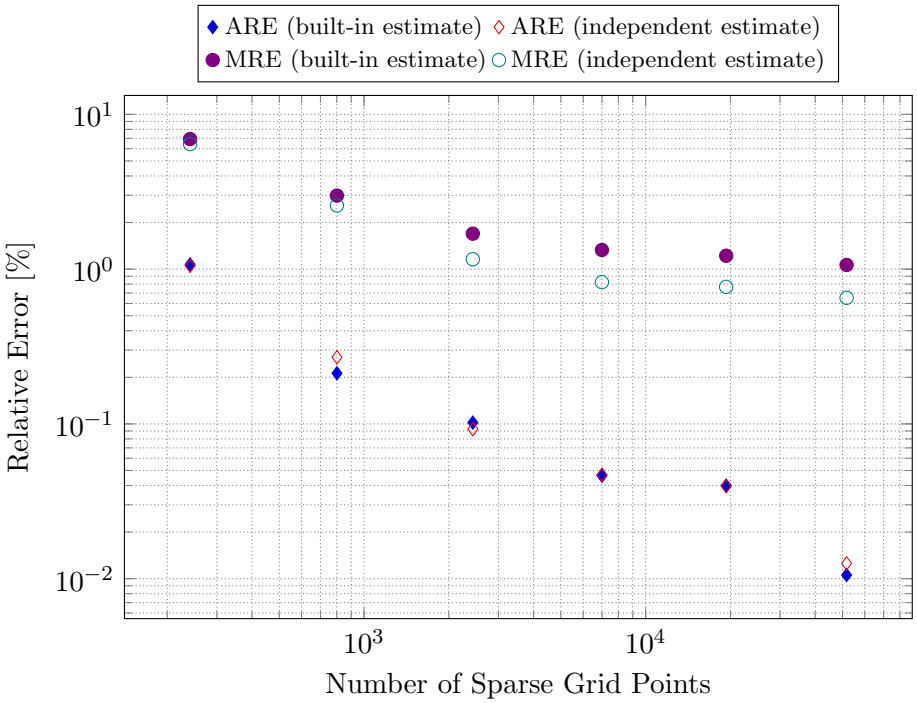


Figure 4.9: PWR assembly: approximation errors in the standard case for the group 6 microscopic absorption cross section of $^{238}_{92}$U

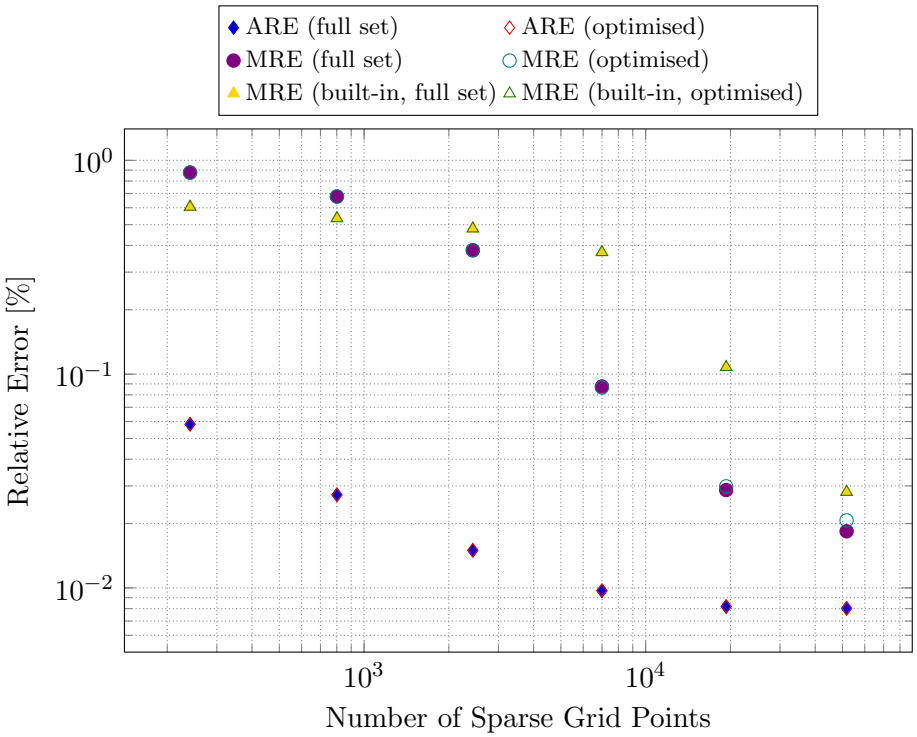


Figure 4.10: PWR assembly: interpolation errors in the standard case for the group 6 microscopic absorption cross section of $^{238}_{92}$U

0.1 % for maximum relative error is achieved at level eight with interpolation. In this case 8192 independent samples were used to estimate ARE and MRE independently.

It is considered illustrative to include the same results for the group six absorption cross section of $^{238}_{92}$U, which is an important cross section for thermal reactors. One can observe from Figure 4.9 that, depending on the sparse grid level, the approximation of this microscopic cross section is between two and six times more accurate than the approximation of the macroscopic absorption cross section. The required average relative error is reached at level eight, with 51713 samples, but the maximum relative error mirrors the slow decay observed in the macroscopic cross section.

The interpolation of this microscopic cross section is also more accurate than that of the macroscopic cross section by a factor of approximately five, as can be seen in Figure 4.10. In this instance both required accuracy levels are reached at level six, with 6993 samples, though the built-in error estimate is so overly conservative that it would only indicate that the maximum relative error requirement had been reached at level seven, with 19319 samples.

The built in error estimate is conservative because it essentially estimates the error of the previous level interpolation at every point that is new to the current level. It is known from the theory that the interpolation error decreases as the sparse grid level increases (Klimke, 2006), therefore the error of the previous level should be greater than the error of the current level. One can clearly see in Figure 4.10 that the error estimated based on the hierarchical surpluses at level seven corresponds very closely to the error estimated on the independent samples at level six. The same phenomenon will repeat itself in most of the interpolation results that will follow.

4.3.3 Accuracy of the representation methods in the MTR example

The same set of calculations as mentioned above were performed for the MTR example and the results for the macroscopic absorption cross section are shown in Figure 4.11 and Figure 4.12. The built-in error estimate for the maximum relative error of the approximation does not match that measured on the independent samples as closely as before, but it is clearly conservative. In this case interpolation does not hold such a clear advantage at lower grid levels, but at levels above five, the interpolation accuracy improves faster than that of the approximation. When approximation is used, neither of the required accuracy levels are achieved at any sparse grid level. When interpolation is used, the required accuracy for both maximum and average relative error is achieved at level six, with 2929 samples.

The group six absorption cross section of $^{238}_{92}$U was also analysed for the MTR example and the results for approximation are presented in Figure 4.13. Once again the maximum relative error decays very slowly and never approaches the limit of 0.1 %, but the average relative error comes reasonably close to the requirement at level six, with 2929 samples.

The results for interpolation of the group six absorption cross section of $^{238}_{92}$U are presented in

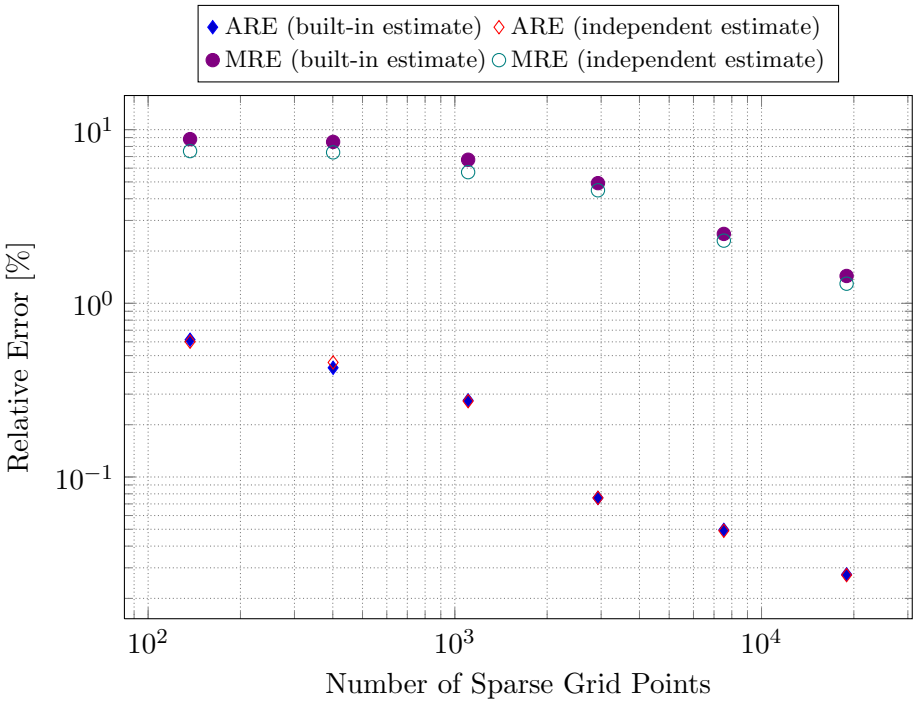


Figure 4.11: MTR assembly: approximation errors in the standard case

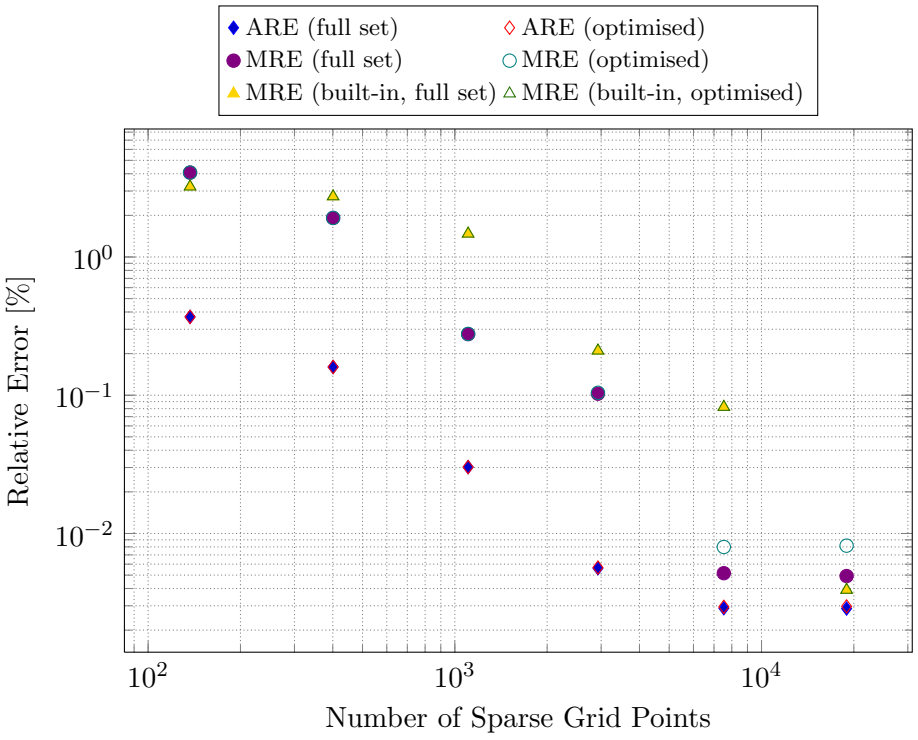


Figure 4.12: MTR assembly: interpolation errors in the standard case

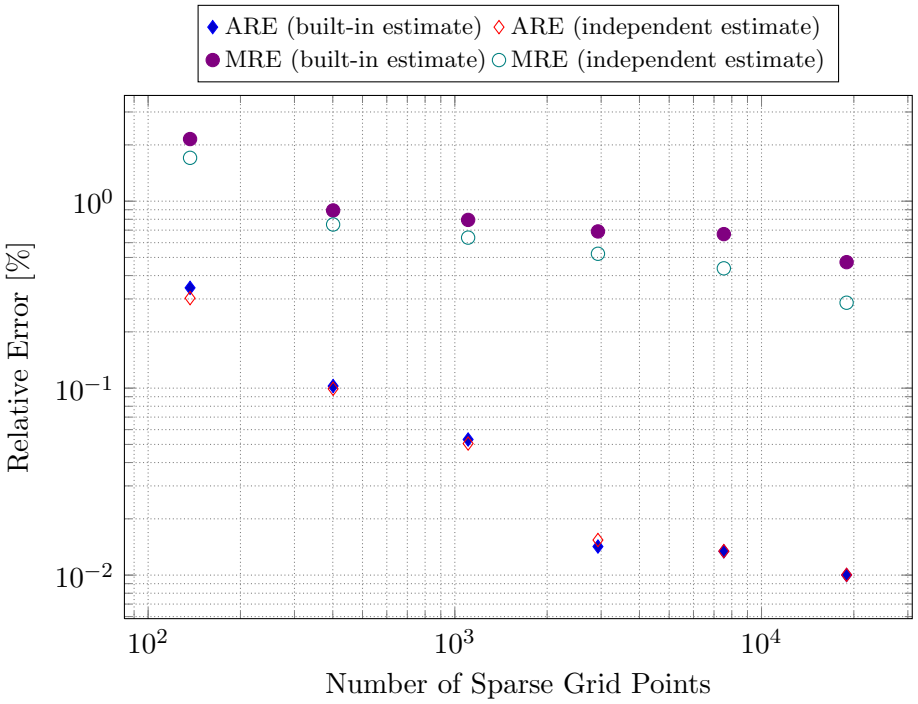


Figure 4.13: MTR assembly: approximation errors in the standard case for the group 6 microscopic absorption cross section of $^{238}_{92}$U

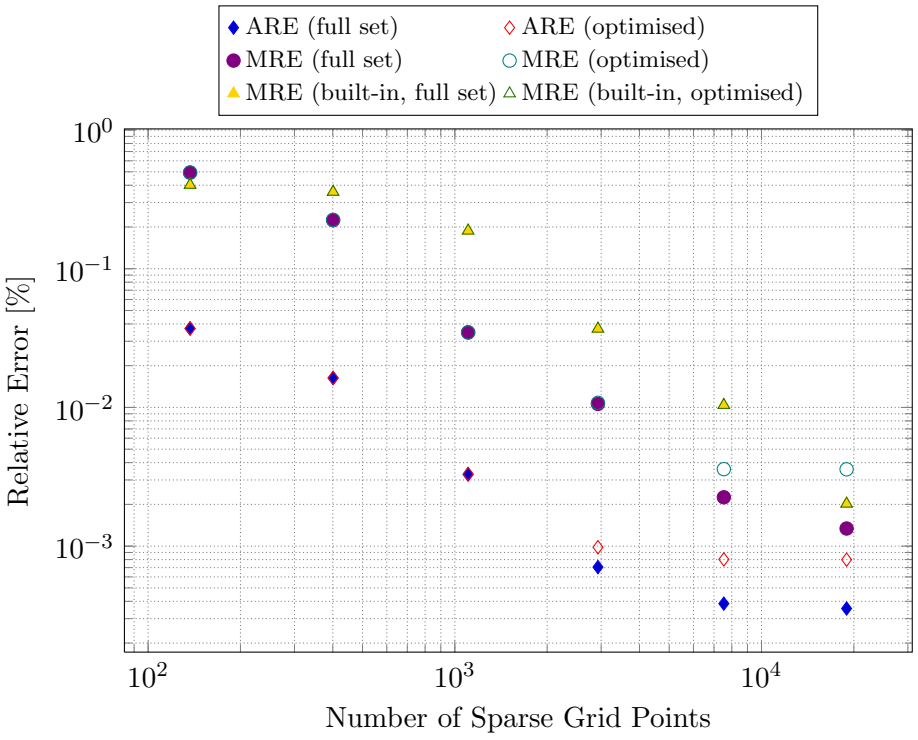


Figure 4.14: MTR assembly: interpolation errors in the standard case for the group 6 microscopic absorption cross section of $^{238}_{92}$U

Figure 4.14. It is clear that, for this cross section, interpolation on a sparse grid can achieve very good accuracy. The interpolation at grid level four already comes close to our accuracy targets, with 401 samples, and the interpolation at at grid level five, with 1105 samples has better accuracy than required. It should, however, be noted that the built-in error estimate is conservative, and although it is reasonably accurate at level four, it is so overly conservative at higher levels that, as was the case for interpolation of this cross section in the PWR example, if one had to rely on the built-in measure, one would only know that the required accuracy had been reached at the next grid level. One may also observe that, for the microscopic cross section, the maximum relative error converges towards the average relative error as localised features are captured by polynomials with small effective support. This behaviour will be observed in other cases as well.

4.3.4 Optimising the representations

The process of optimising the representation by rejecting terms, that make a small contribution or are poorly estimated, as is explained in Section 4.2.1, was applied to the representations of the cross section dependence of both the MTR and PWR assemblies.

It should be noted that the optimisation of the approximation is done before the library is created (to minimise storage space) and the post-representation processing is performed at a later stage, after reading the library. Information on the accuracy of the approximation before the optimisation is performed is therefore not available. It is expected that this accuracy would be worse than the accuracy of the optimised library, due to the error that may be introduced by poorly estimated terms.

In the case of interpolation, no library is created or stored on the hard drive and the post-representation processing is performed directly after the interpolation is constructed, therefore post-representation processing can be done for the interpolation before and after optimisation separately. Figure 4.8 and Figure 4.12 indicate that, at least in the case of interpolation, the truncation error, that results from the optimisation, is very small. In fact, in the case of interpolation, the markers of the errors before and after the optimisation coincide so closely that in many instances they can not be distinguished from each other on the scale to which these graphs are drawn.

In order to gain a feeling of the impact of optimisation on the size of the representation, Table 4.5 shows the characteristics of the optimised approximation, such as the number of terms that are retained and the maximum polynomial order of the final approximation. The number of terms that are initially in the interpolation is equal to the number of samples that were used to construct it, as every basis function is associated with a particular node. The number of terms that were retained in the optimised library can bee seen in Table 4.6.

It is clear that, at least in the case of interpolation, a large number of terms are retained even after optimising the representation, and this is still insufficient for an accurate representation of the

macroscopic cross section. This is partly due to the complexity of the macroscopic cross section dependencies. The optimised representation of the $^{238}_{92}$U microscopic cross section only requires 764 terms to achieve the required average relative error, while 1124 terms are still insufficient to reach this accuracy for the macroscopic cross section in the PWR representation.

4.3.5 Possible cause of inaccuracy in the representations

It is noticeable in both the PWR example, and the MTR example that the maximum relative error of the approximation decays very slowly. Some further investigation showed that, in the MTR example, at grid level eight, increasing the limit of the maximum one-dimensional polynomial order to $B_\infty = 35$ instead of $B_\infty = 25$ significantly decreased the maximum relative error of the approximation from 1.3 % to 0.61 %. Close inspection of the library revealed that the high order polynomials were only used in burnup, and that the very high order polynomials shared no cross terms with the other state parameters.

Previous experience at Necsa with MTR and PWR cross section dependencies indicated that a

Table 4.5: Number of approximation terms after the optimisation and the highest order of monomial that was retained. The MTR case initially had 11100 terms and the PWR case 13400 terms

	MTR		PWR	
level	final number of terms	highest monomial order	final number of terms	highest monomial order
3	4	1	7	1
4	8	1	16	3
5	14	4	23	6
6	25	24	51	6
7	41	25	77	17
8	51	25	108	25

Table 4.6: Number of interpolation terms before and after optimisation

	MTR		PWR	
level	initial number of terms	final number of terms	initial number of terms	final number of terms
3	137	69	241	191
4	401	128	801	421
5	1105	195	2433	606
6	2929	263	6993	765
7	7537	334	19319	962
8	18945	354	51713	1146

major contributor to the error may be the brief Xenon transient at BOL (beginning of life). This supposition was confirmed by projecting the error of the level eight MTR approximation with a maximum polynomial order of $B_\infty = 25$ from each of the 4096 independent test samples onto the state parameters, one at a time. The result of this procedure is displayed in Figure 4.15. The same was done for the PWR assembly to confirm that similar behaviour is exhibited at BOL.

It is clear from the figure that the bulk of the error is concentrated at the beginning of life and that the state parameter that drives this error is burnup. This analysis also offers an explanation as to why interpolation is so much more accurate than approximation: the localised nature of the basis functions makes it easier for the interpolation to capture complicated behaviour on a small part of the domain. It is also interesting to note that the spike in error at BOL can be observed for various microscopic cross sections as well as for the macroscopic cross section demonstrated in Figure 4.15.

4.4 Setting Xenon Concentration to its Equilibrium Value

In order to confirm that the large error at the beginning of life could really be attributed to Xenon, the calculations from the standard case were repeated, but the Xenon number density was set to its equilibrium value throughout the calculation.

4.4.1 Accuracy of the representation methods in the PWR example

Figure 4.16 shows that, in the case of the PWR example, there is little impact on the accuracy of the approximation at low levels. The rate at which the maximum relative error decays is, however, greater with Xenon number density set to its equilibrium value than in the standard case, therefore leading to a clear improvement in maximum relative error. The improvement in the average relative error over that of the standard case is small, as may be expected. The Xenon transient is very localised in the problem domain, and therefore makes only a small contribution to the average error, even though the maximum error is very large. Neither of the accuracy targets are reached at any grid level.

Figure 4.17 shows an improvement in the accuracy of the interpolation at low grid levels and in the rate at which the maximum relative error decays. It also shows that both the maximum and the average relative error do not decrease after level five. This confirms the supposition that the Xenon transient is the source of the largest errors for both approximation and interpolation, but also indicates that there are other sources of error that set a limit to the representation accuracy. Only the target for the maximum relative error is ever reached, at grid level five, with 2433 samples.

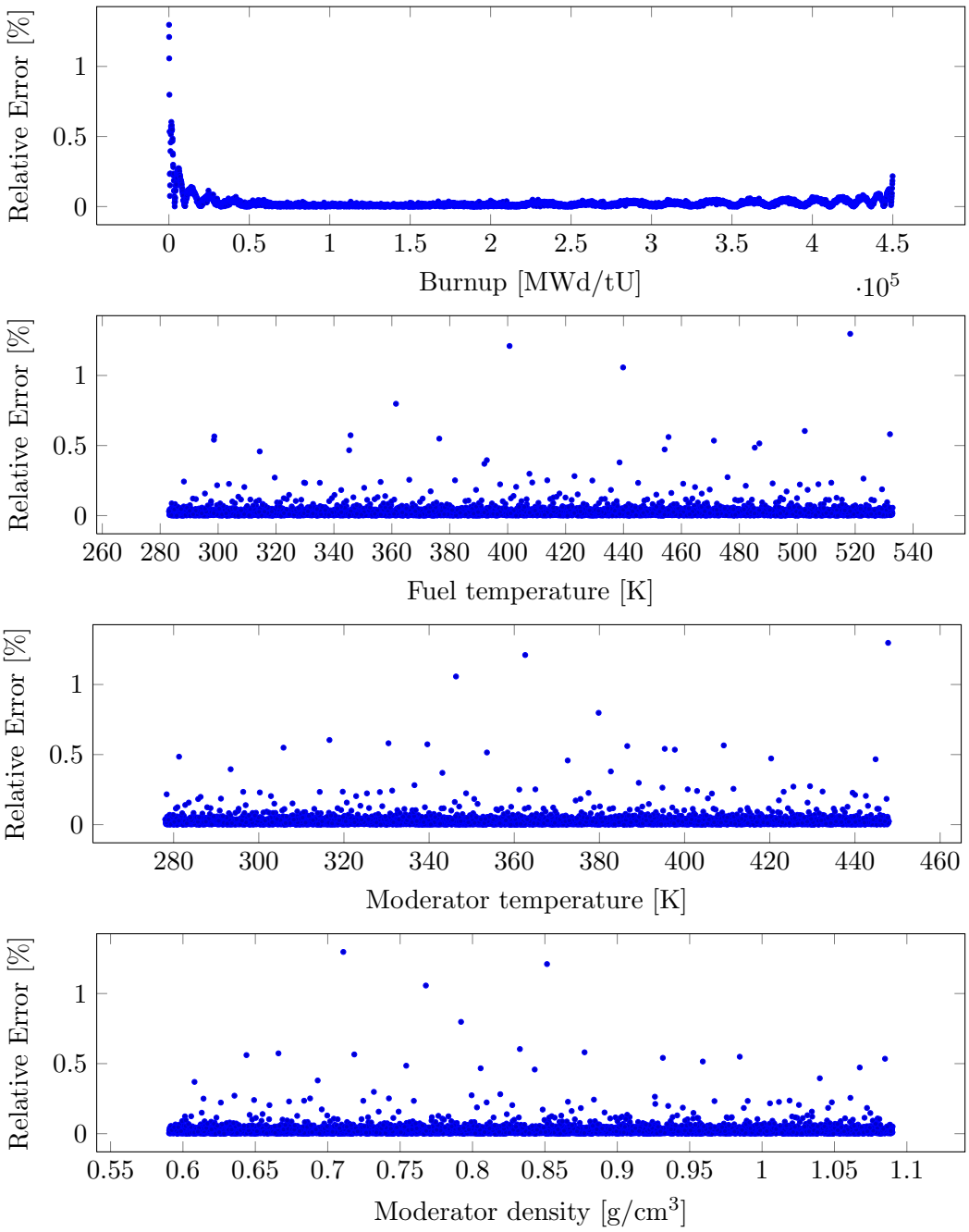


Figure 4.15: Relative error of the approximation, constructed using a sparse grid of level eight and a maximum polynomial order of 25, at each of the independent test samples, projected onto the individual state parameters for the MTR example

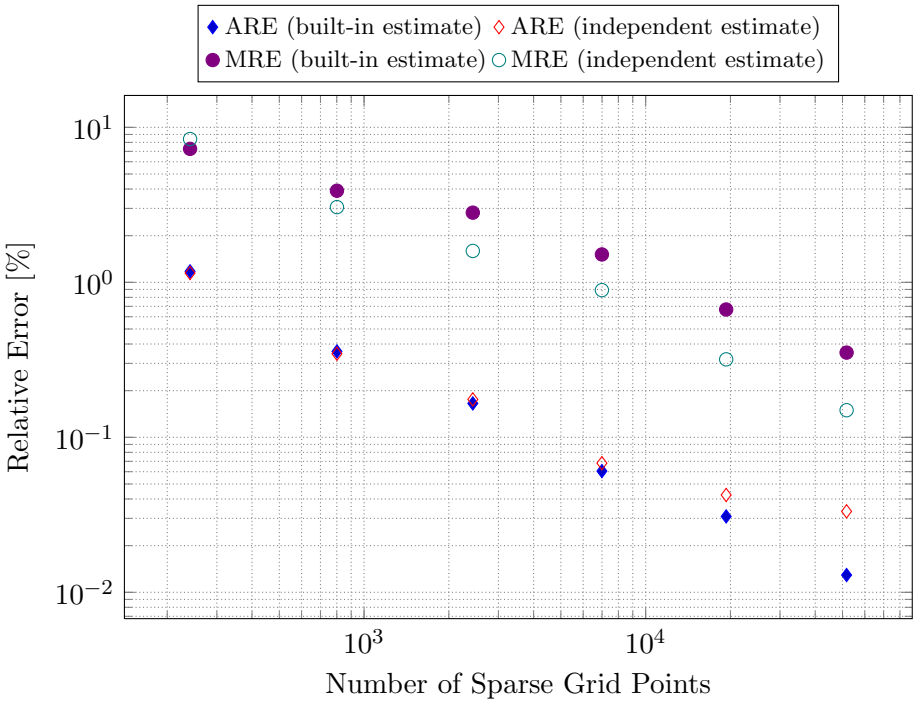


Figure 4.16: PWR assembly: approximation errors with Xenon number density set to its equilibrium value

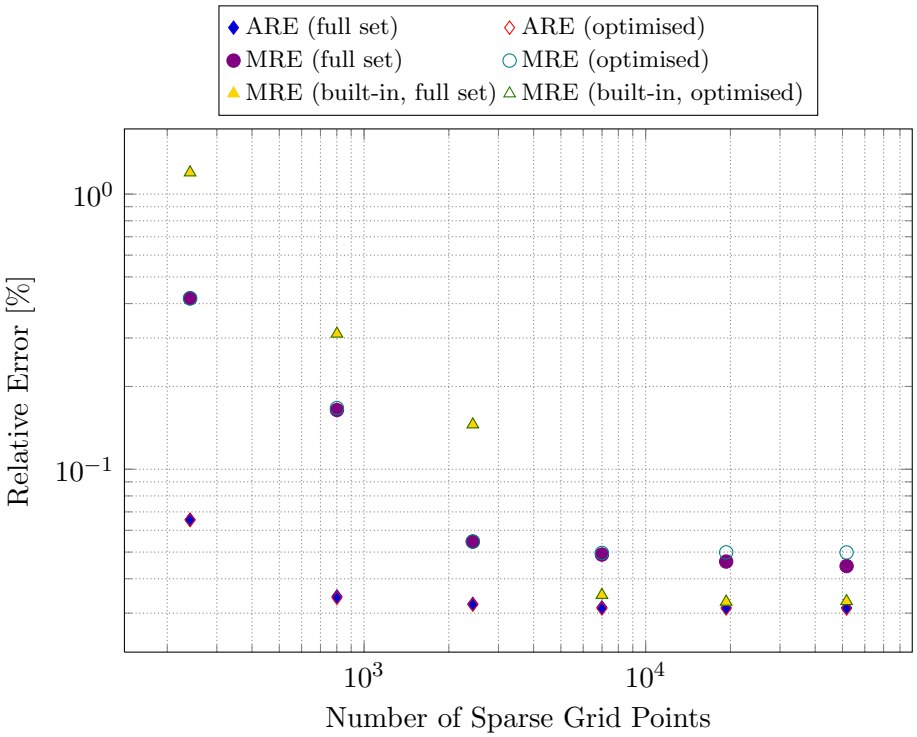


Figure 4.17: PWR assembly: interpolation errors with Xenon number density set to its equilibrium value

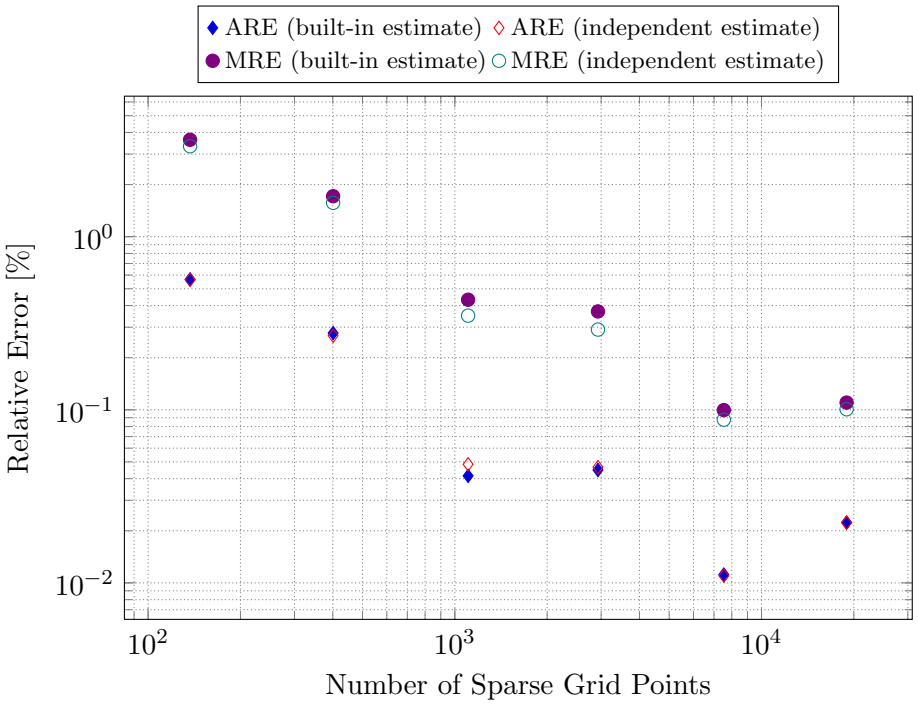


Figure 4.18: MTR assembly: approximation errors with Xenon number density set to its equilibrium value

4.4.2 Accuracy of the representation methods in the MTR example

From Figure 4.18 and Figure 4.19 one can observe that the small change of eliminating the Xenon transient at BOL has increased the accuracy by almost an order of magnitude, when compared to the standard case, for both the approximation and the interpolation in the MTR example. The approximation now achieves the stated acceptable accuracy for both maximum and average relative error, albeit only at level seven with 7537 samples. The interpolation reaches the target accuracy for both maximum and average relative error by level four, which contains 401 samples.

An MTR assembly undergoes many cycles and the Xenon concentration is therefore rarely at equilibrium. It would be preferable to add Xenon concentration as an extra state parameter, but this work could not be completed in time to be included in this dissertation. This addition is actively being pursued and its inclusion in this cross section representation framework is intended as future work.

4.4.3 Accuracy of the representation methods in the VVER example

In addition to the MTR and PWR examples, for which several other cases were investigated, the VVER example was used for this case where Xenon concentration starts at its equilibrium value. It is an interesting example to compare the others with, given the differences between them. The

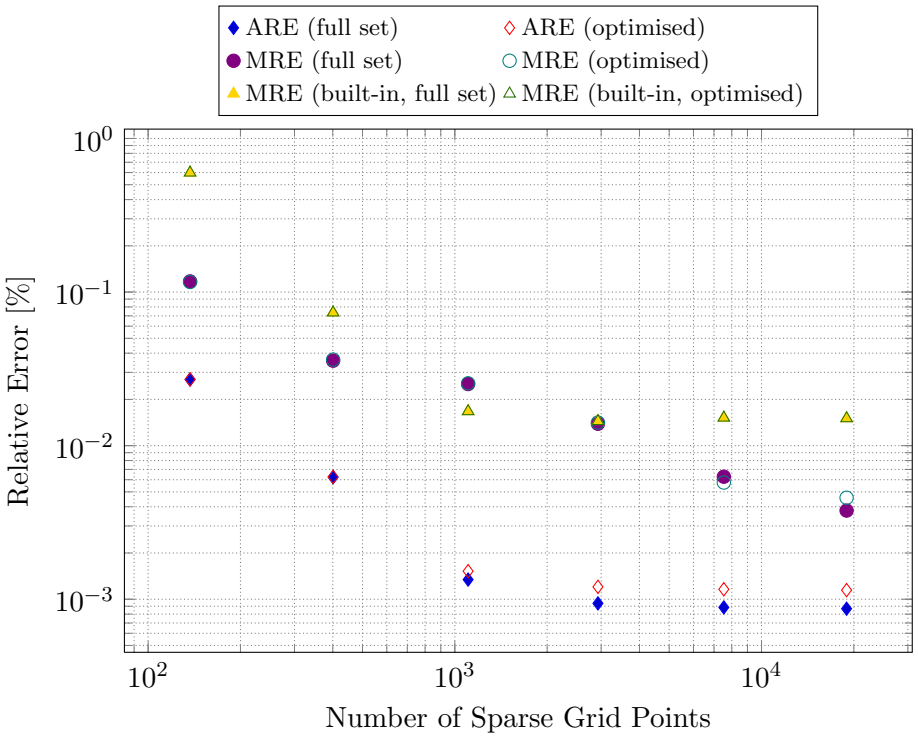


Figure 4.19: MTR assembly: interpolation errors with Xenon number density set to its equilibrium value

VVER example is a single pin, as opposed to a full assembly, has a different group structure and was calculated by a different transport code. The results for interpolation of the thermal absorption cross section are presented in Figure 4.20 for grid levels one to six.

It can be seen that the target accuracy is achieved at grid level four, which has 801 points, though the slightly conservative built-in estimate would only indicate so at level five, which has 2433 points. Also note that at grid level six the maximum and average relative error converge to an extreme accuracy of 2×10^{-3} %.

4.4.4 Summary remarks

Given the results of Section 4.3 and Section 4.4, some improvements to the cross section representation procedure were attempted. One improvement was splitting the problem into two separate libraries, a so-called *start-up* library, which covered the first part of the burnup interval until Xenon had reached a stable concentration, and a so-called *operational library*, which covered the rest of the element's lifetime. Another improvement was to use an anisotropic sparse grid to sample the cross sections, as described in Section 2.3.2. It is commonly known and confirmed by this study that, for Light Water Reactors, the primary dependence is on burnup, and it would therefore be preferable to have more samples in the burnup dimension than in the others.

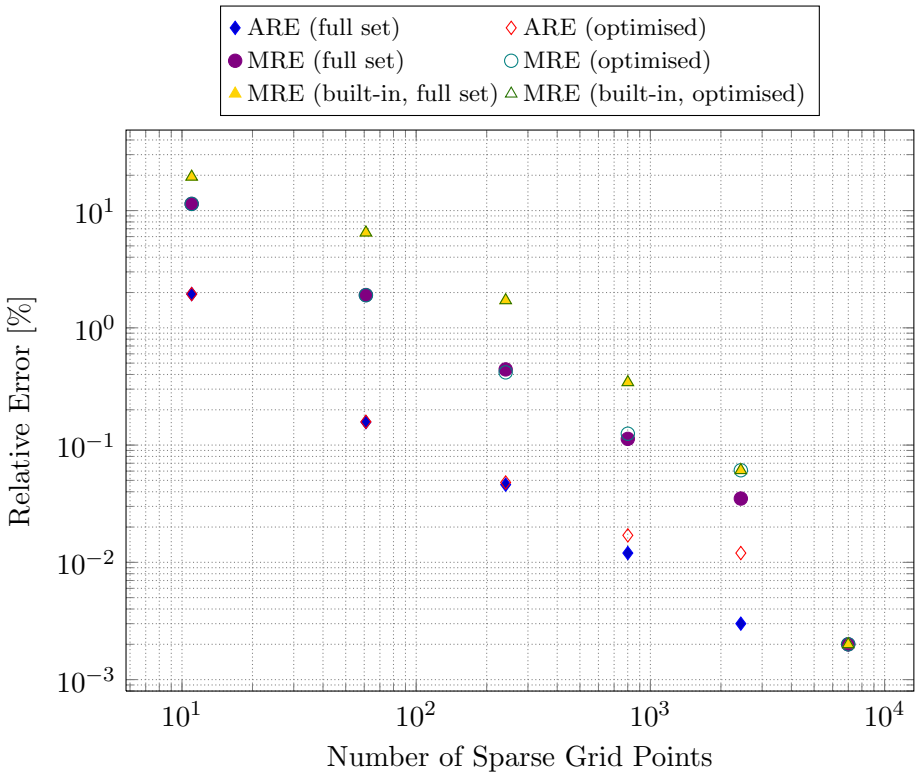


Figure 4.20: VVER fuel pin: interpolation errors with Xenon number density set to its equilibrium value

4.5 Splitting the Burnup Interval

After ascertaining that the Xenon transient is the major contributing factor to the error of representation, it was decided to split the burnup interval and thereby create two separate libraries: a start-up library and an operational library. The idea behind this improvement is that behaviour that is localised when considered over the whole burnup interval should be easier to represent on a smaller interval.

Two separate calculations were done to make the two libraries. A representation was constructed on the first section of the burnup interval, with the other state parameters retaining their intervals of variation and nominal values as in Table 4.2 and Table 4.3. Another representation was constructed independently on the remainder of the burnup interval, also without modification of the rest of the state parameter space. Samples that correspond to a level seven sparse grid were calculated on each interval, separately. In order to sample from the sparse grid on the remainder of the interval, the feature of being able to specify a separate set of burnup points, that are not employed for the representation, was used to burn through the initial burnup interval. The nestedness of the sparse grid was again utilised to create representations on grids of levels less than seven. The representation error was estimated during the post-representation processing using 4096 quasi-random samples in each of the two sections of the domain.

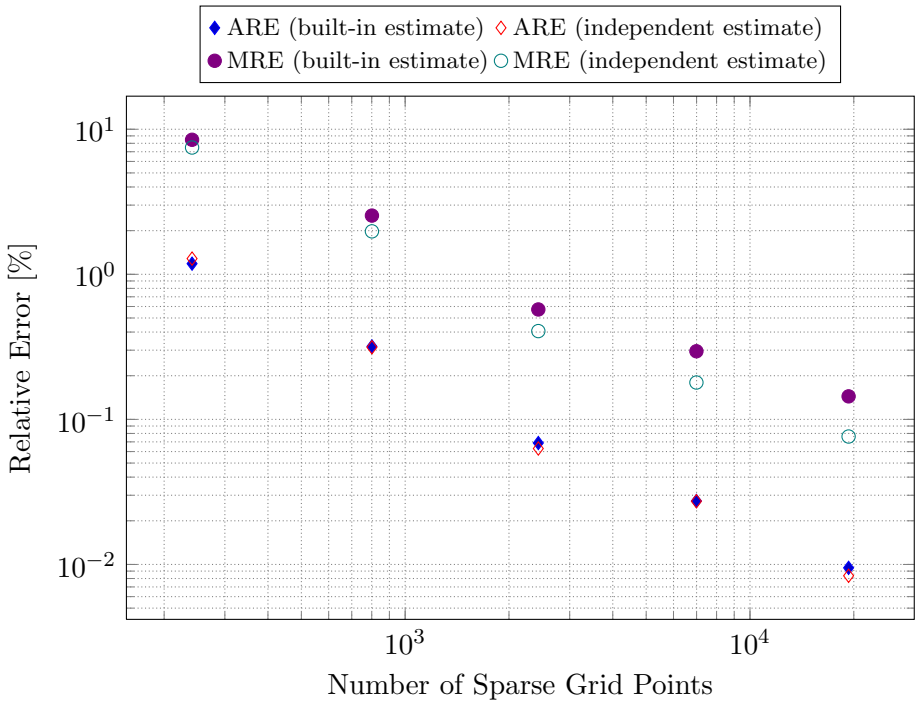


Figure 4.21: PWR assembly: approximation errors on the burnup interval from 0 to 100 MWd/tU

4.5.1 Accuracy of representation for the PWR example

The full burnup interval of the PWR example is from 0 to 60000 MWd/tU. By plotting finely spaced samples of the macroscopic absorption cross section as a function of burnup with all other state parameters set to their nominal values, it was ascertained that the Xenon transient lasts for approximately the first 100 MWd/tU. This corresponds to approximately 2.4 days at full power.

The accuracy of the approximation for the first 100 MWd/tU is shown in Figure 4.21. It is clear that the desired accuracy is only achieved at level seven, which contains 19319 samples. The accuracy of the interpolation, however, reaches the stated targets at grid level four, with 801 samples, as can be observed from Figure 4.22.

One can see from Figure 4.23 that the approximation through the remainder of the burnup interval shows a small improvement in the average relative error, as compared to the standard case, but still does not reach the target accuracy at any grid level. This is to be expected and can be attributed to the short duration of the Xenon transient. Average error will not be influenced by large errors in a very small part of the domain, therefore it was expected that average relative error would be similar to the standard case. The maximum relative error decays faster, but also does not reach the target accuracy at any grid level.

The interpolation of the remainder of the burnup interval shows some improvement in the average relative error over the standard case at levels three and four, but the average relative error does not

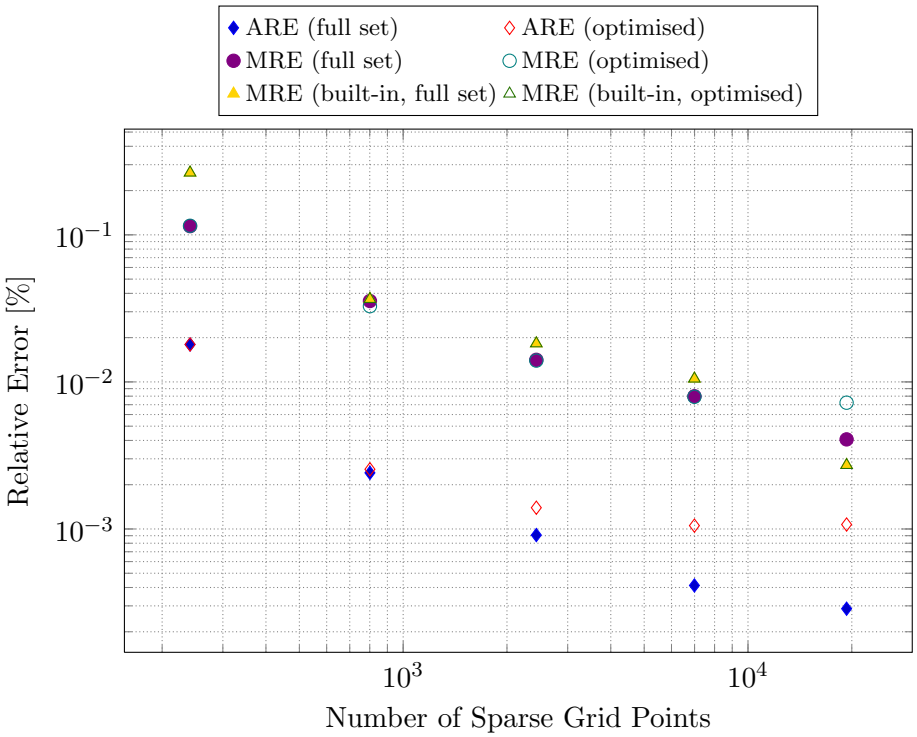


Figure 4.22: PWR assembly: interpolation errors on the burnup interval from 0 to 100 MWd/tU

decay after these initial two levels, as can be seen from Figure 4.23, and never reaches the target accuracy. The maximum relative error decays well initially, and reaches the target accuracy at level four, with 801 samples, but does not improve at levels six or seven. For the PWR example, therefore, splitting the burnup interval does improve the accuracy of representation significantly at the beginning of the cycle, but not through the rest of the cycle.

The lack of improvement of results for the representation on the 100 to 60000 MWd/tU burnup interval may become clearer in the context of Figure 4.25. This shows that the error at higher depletion may be driven by the other state variables. Indeed, inspection of the approximation library shows that, in addition to burnup, which has a maximum polynomial order of 11, moderator temperature has a maximum polynomial order of 5. This indicates that the dependence of the cross sections on moderator temperature may be a source of complexity and therefore inaccuracy in the representation. Alternatively, the dependence on burnup may become more complex with increasing exposure. A full analysis of variance would therefore need to be carried out for the PWR example, and an appropriate representation method chosen according to that analysis.

Even if one were to use the more effective method, interpolation, one would need a total of 1602 samples to create a combined library with a maximum relative error of 0.1 % across the full burnup interval and this library would be deficient if the requirement on the average relative error was deemed important. Splitting the burnup interval is therefore not a good solution for creating a representation of PWR cross sections on a large domain.

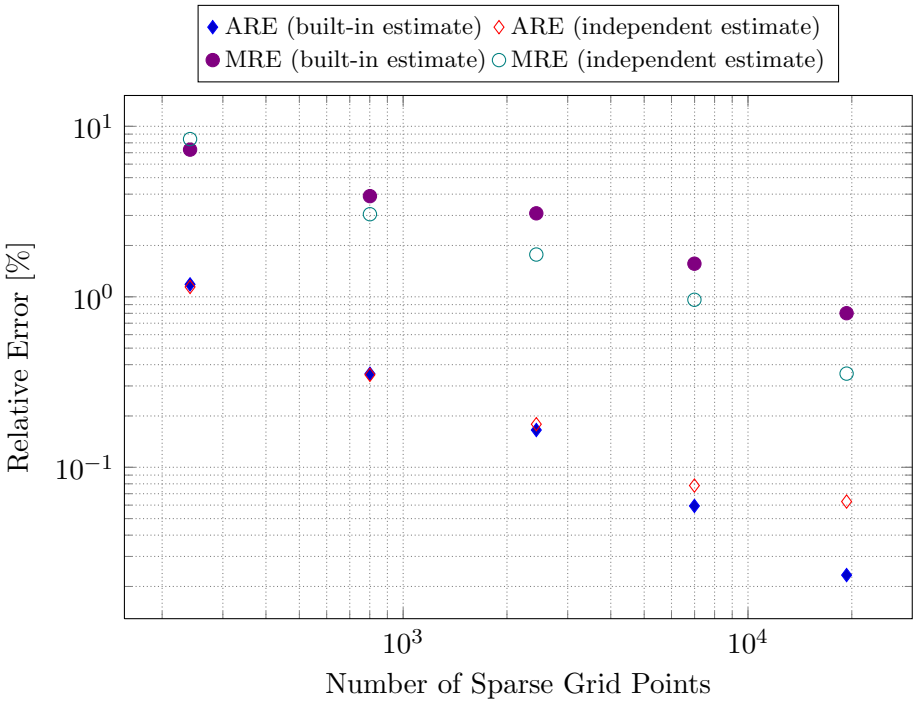


Figure 4.23: PWR assembly: approximation errors on the burnup interval from 100 to 60000 MWd/tU

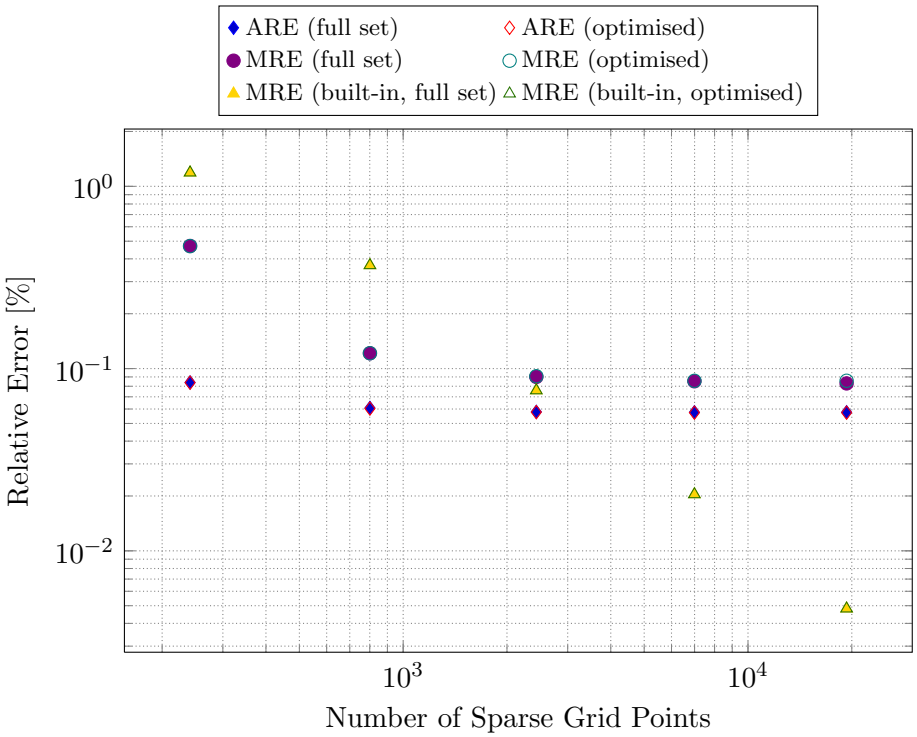


Figure 4.24: PWR assembly: interpolation errors on the burnup interval from 100 to 60000 MWd/tU

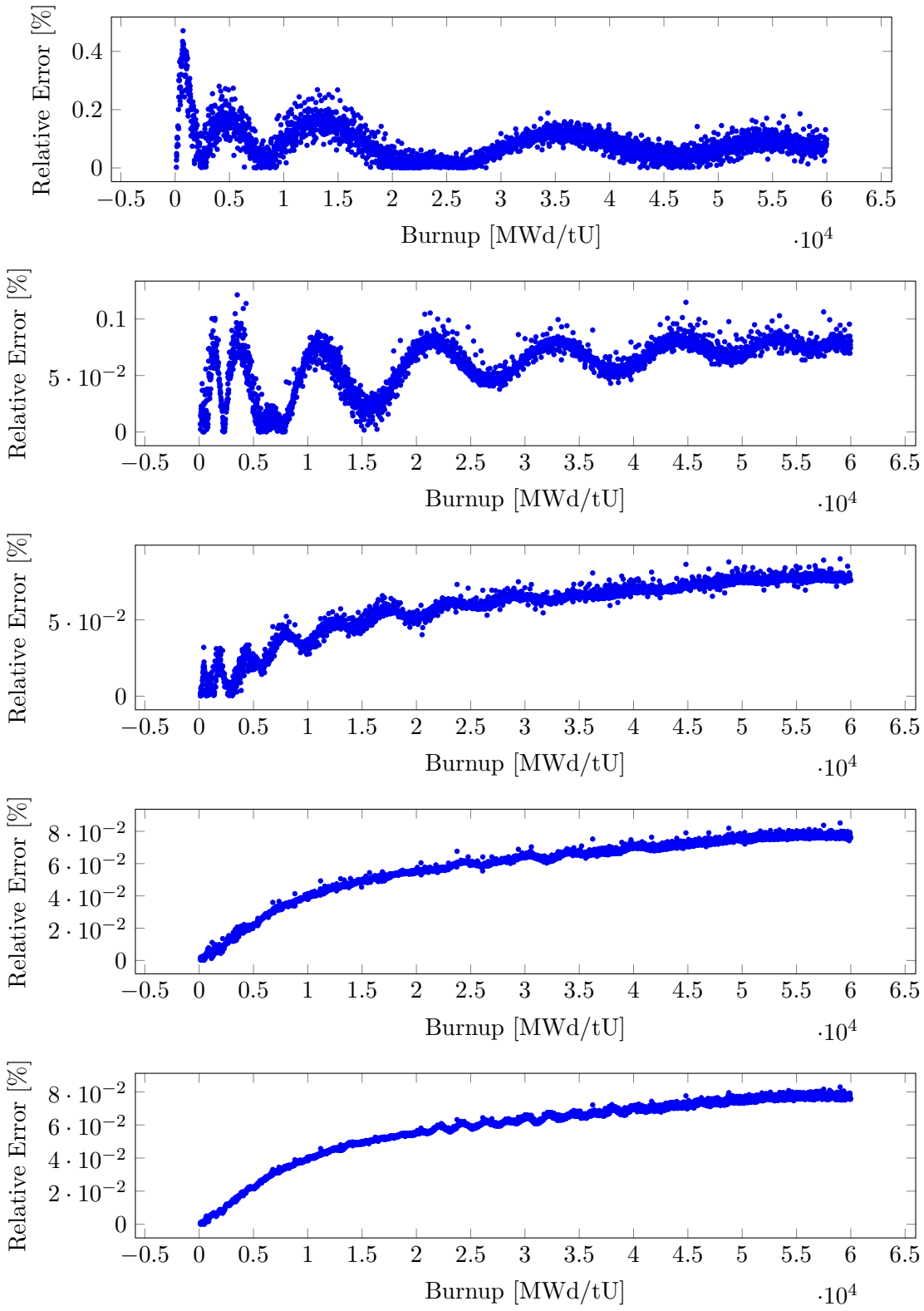


Figure 4.25: Relative error of the interpolation on the interval 100 MWd/tU to 60000 MWd/tU as a function of burnup for levels from three (at the top) to seven (at the bottom)

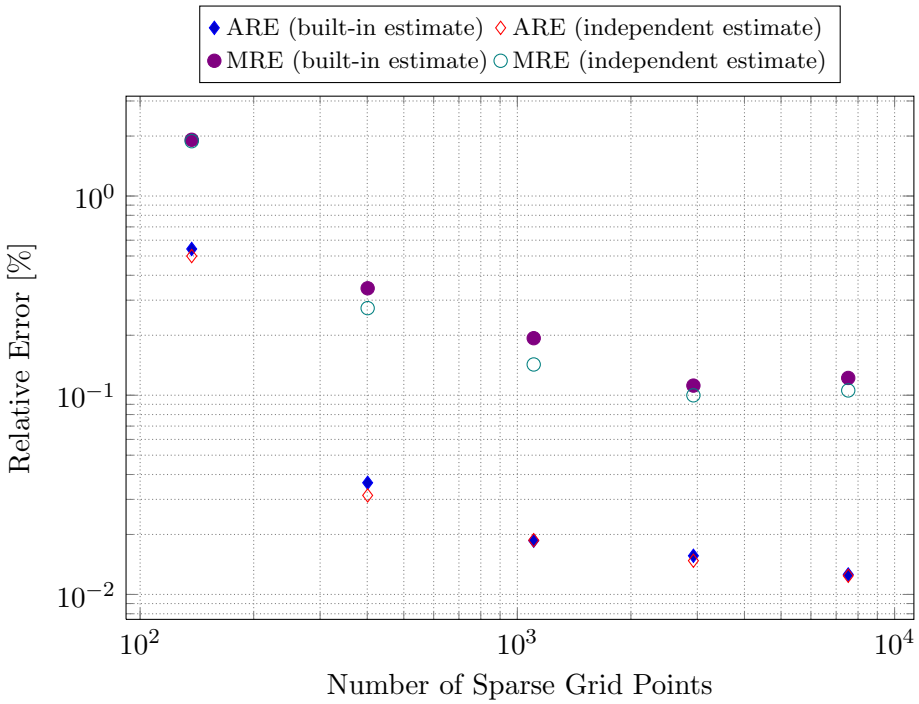


Figure 4.26: MTR assembly: approximation errors on the burnup interval from 0 to 3000 MWd/tU

4.5.2 Accuracy of representation for the MTR example

The full burnup interval of the MTR example is from 0 to 450000 MWd/tU. By plotting finely spaced samples of the macroscopic absorption cross section as a function of burnup with all other state parameters set to their nominal values, it was ascertained that the Xenon transient lasts for about the first 3000 MWd/tU. This corresponds to approximately 1.5 days at full power.

Figure 4.26 shows that, on this small burnup interval, approximation requires 2929 samples to reach the target accuracy. Figure 4.27 shows that interpolation exceeds the target for the maximum relative error and matches the target for the average relative error with 137 samples.

On the rest of the burnup interval, from 3000 to 450000 MWd/tU, approximation does not reach the target accuracy at any grid level, as shown in Figure 4.28. Interpolation, however, reaches the desired accuracy at grid level four, with 401 samples, as can be seen in Figure 4.29. This means that one can create a combined library that adheres to the stated accuracy requirements with $137 + 401 = 538$ samples, and the built-in error estimates will be reliable for this library. Splitting the burnup interval and using sparse grid interpolation may therefore be a viable solution for creating an accurate MTR library.

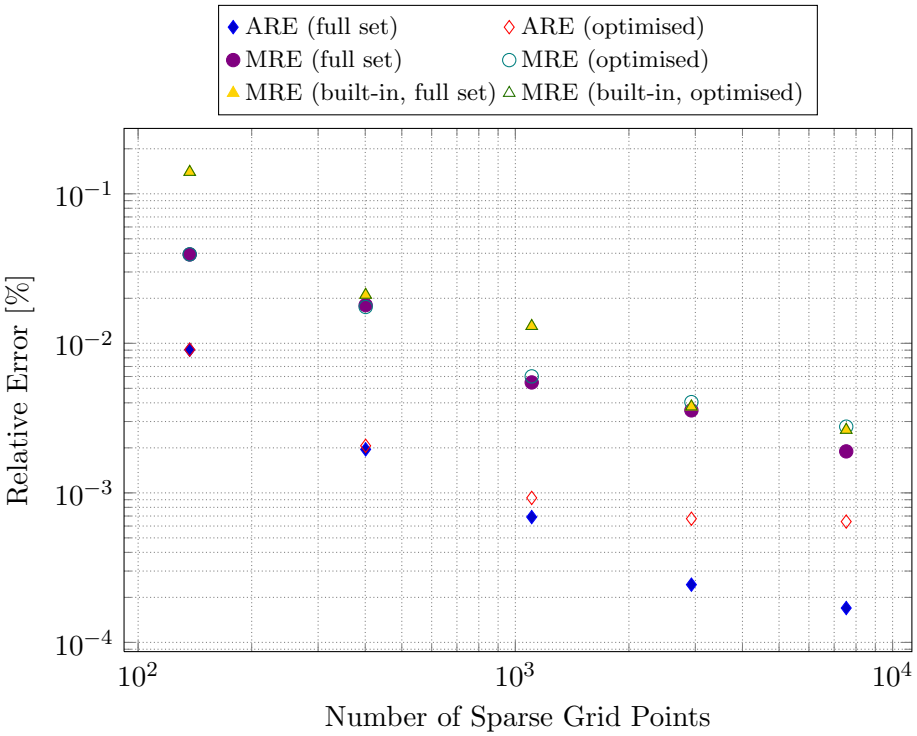


Figure 4.27: MTR assembly: interpolation errors on the burnup interval from 0 to 3000 MWd/tU

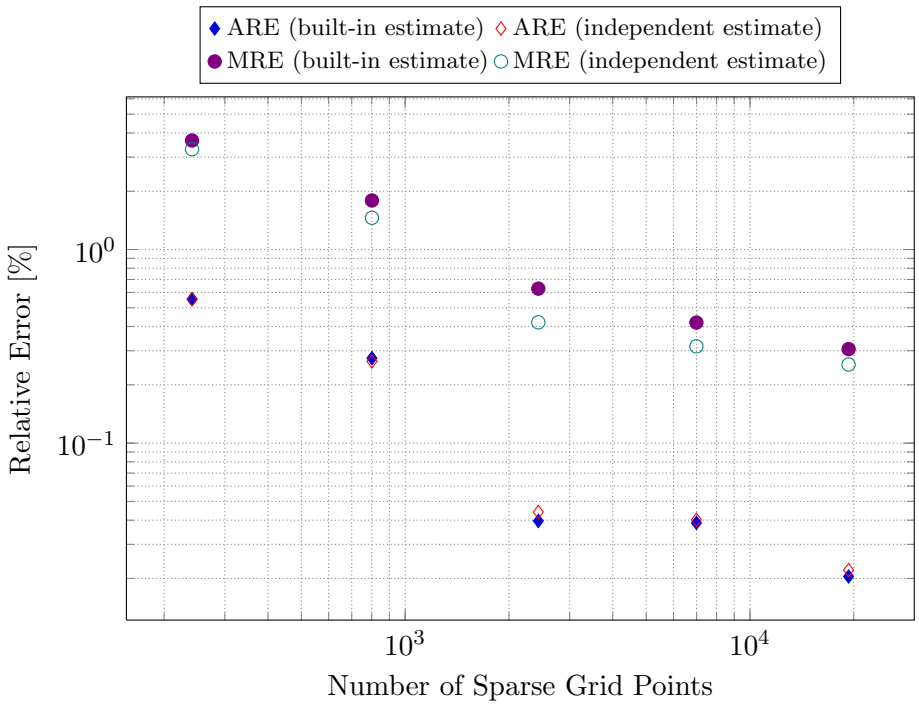


Figure 4.28: MTR assembly: approximation errors on the burnup interval from 3000 to 450000 MWd/tU

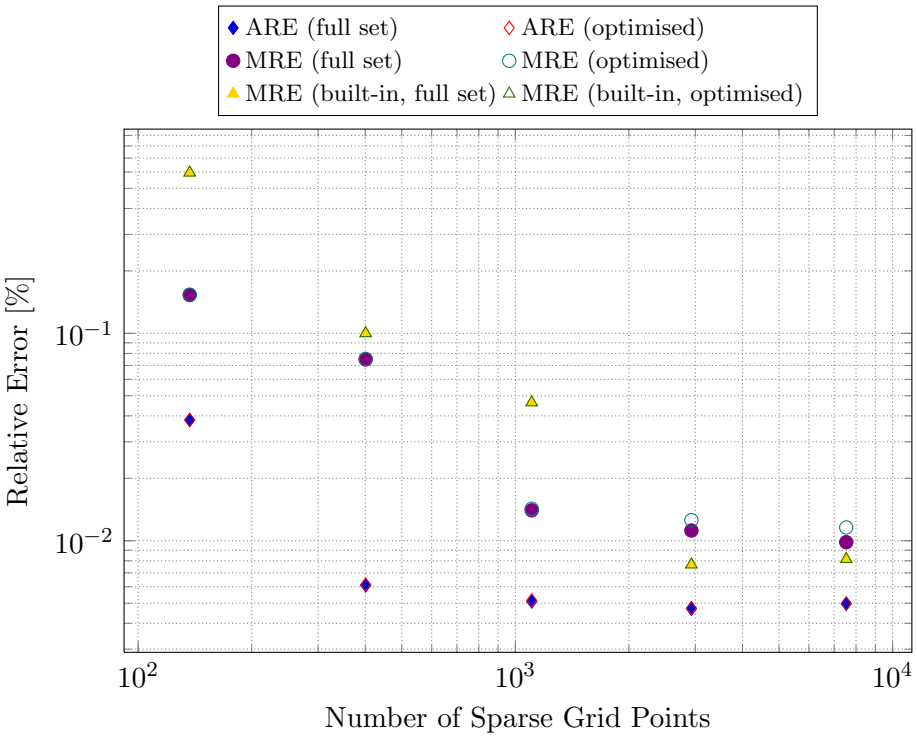


Figure 4.29: MTR assembly: interpolation errors on the burnup interval from 3000 to 450000 MWd/tU

4.6 Anisotropic Sparse Grid Sampling

The anisotropy vector was chosen in all cases as 1 in the burnup dimension and 2 in all other dimensions, i.e. $\boldsymbol{\alpha} = (1, 2, 2, \ldots)$. The number of samples in an anisotropic grid with this anisotropy vector at different levels for 4 and 5 dimensional grids can be seen in Table 4.7.

4.6.1 Accuracy of approximation

Approximation was performed, using this anisotropy vector, for both the MTR and the PWR examples. In both cases the decay of all error measures, both built-in and independently estimated, was very small. In some cases the accuracy decreased with an increase in the number of samples. No graphs of the error as a function of the number of sparse grid samples will therefore be shown, as approximation on a sparse grid with this anisotropy vector can be discarded as a viable representation method out of hand. This may be due to an inappropriate choice of anisotropy vector. Future work will be done to establish the effect that using different anisotropy vectors has on the accuracy of approximation. Interpolation on an anisotropic sparse grid with this anisotropy vector, on the other hand, showed very promising results, which will be discussed below.

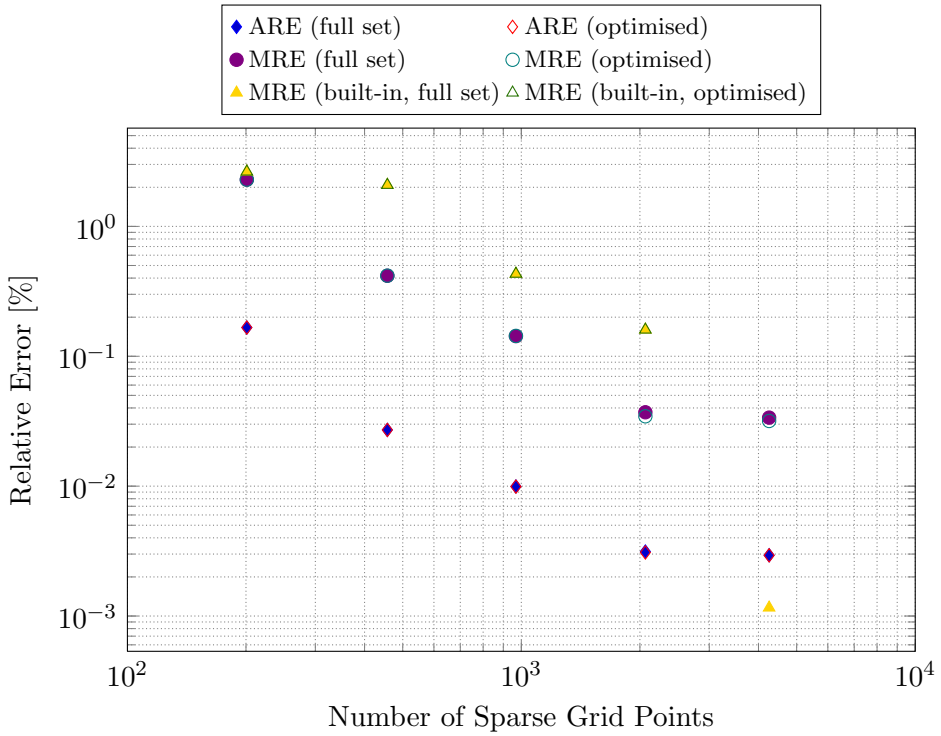


Figure 4.30: PWR assembly: interpolation errors on an anisotropic grid with $\boldsymbol{\alpha} = (1, 2, 2, 2, 2)$

4.6.2 Accuracy of interpolation for the PWR example

The results for interpolation of the macroscopic absorption cross section in the PWR example, using an anisotropy vector of $\boldsymbol{\alpha} = (1, 2, 2, 2, 2)$, are shown in Figure 4.30. Because the relation between sparse grid level and the number of points in the grid is different for anisotropic grids than for isotropic grids, the figure shows results for grid levels from five to nine. One can observe that the target accuracy is achieved at grid level seven, with 969 samples, and exceeded at grid level eight, with 2065 samples, but that the conservative error estimate would not indicate success at these levels. Increasing the grid level further does not seem to improve accuracy and it can be seen that the built-in error measure fails at grid level nine.

For the sake of interest, the interpolation errors for the group six microscopic $^{238}_{92}$U absorption cross section is shown in Figure 4.31 for grid levels from five to nine. In this case, target accuracy is

Table 4.7: Number of sparse grid points in an anisotropic grid with $\boldsymbol{\alpha} = (1, 2, 2, \ldots)$ in 4 and 5 dimensions

d \ q	4	5	6	7	8	9	10	11	12
4	65	141	301	621	1281	2601	5289	10665	21521
5	89	201	457	969	2065	4257	8817	17937	36593

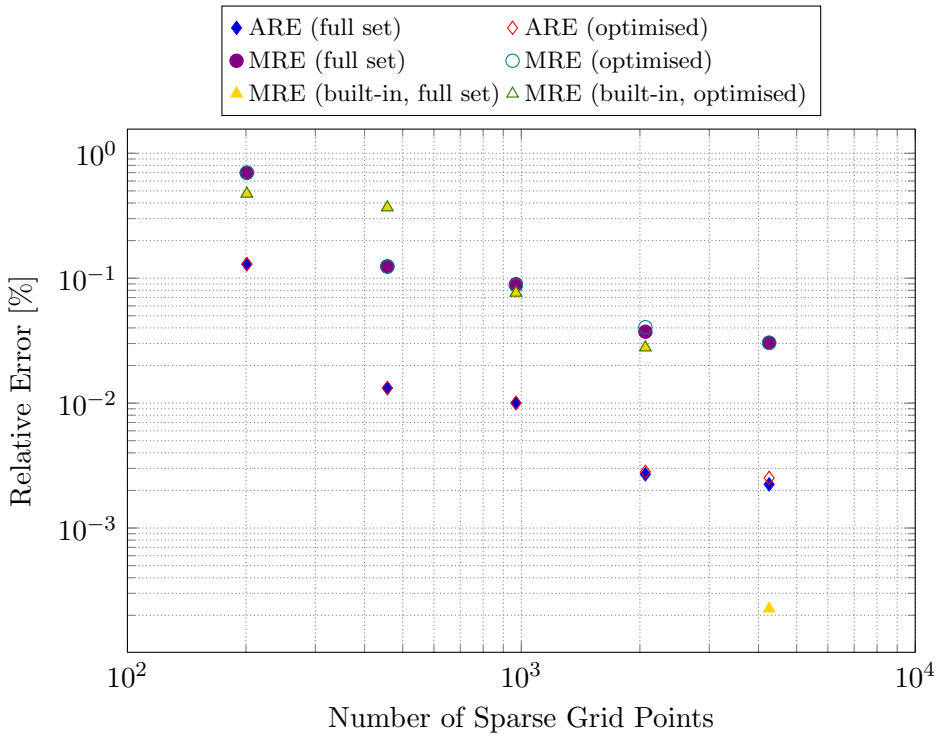


Figure 4.31: PWR assembly: interpolation errors of the $^{238}_{92}$U absorption cross section on an anisotropic grid with $\boldsymbol{\alpha} = (1, 2, 2, 2, 2)$

reached at level six, with 457 samples, but the built-in error measure again fails at grid level nine. It is clear that sparse grid interpolation on an anisotropic grid with $\boldsymbol{\alpha} = (1, 2, 2, 2, 2)$ is a strong contender as a method for constructing a representation of PWR cross sections on a large domain, both for macroscopic and microscopic cross sections.

4.6.3 Accuracy of interpolation for the MTR example

Figure 4.32 shows the accuracy improvement with increasing grid level on an anisotropic grid with anisotropy vector $\boldsymbol{\alpha} = (1, 2, 2, 2)$ for the group six macroscopic absorption cross section of the MTR example at grid levels from four to nine. The target accuracy is reached at level six, with 301 samples, making this anisotropic grid even more suited to representing MTR cross sections than the split burnup approach discussed before.

4.7 Discussion

A review of the results that were presented above may lead one to conclude that sparse grid representation of few group cross sections is not an accurate or successful method for PWR assemblies. It is claimed that a level eight grid is used only for illustration, but in the standard

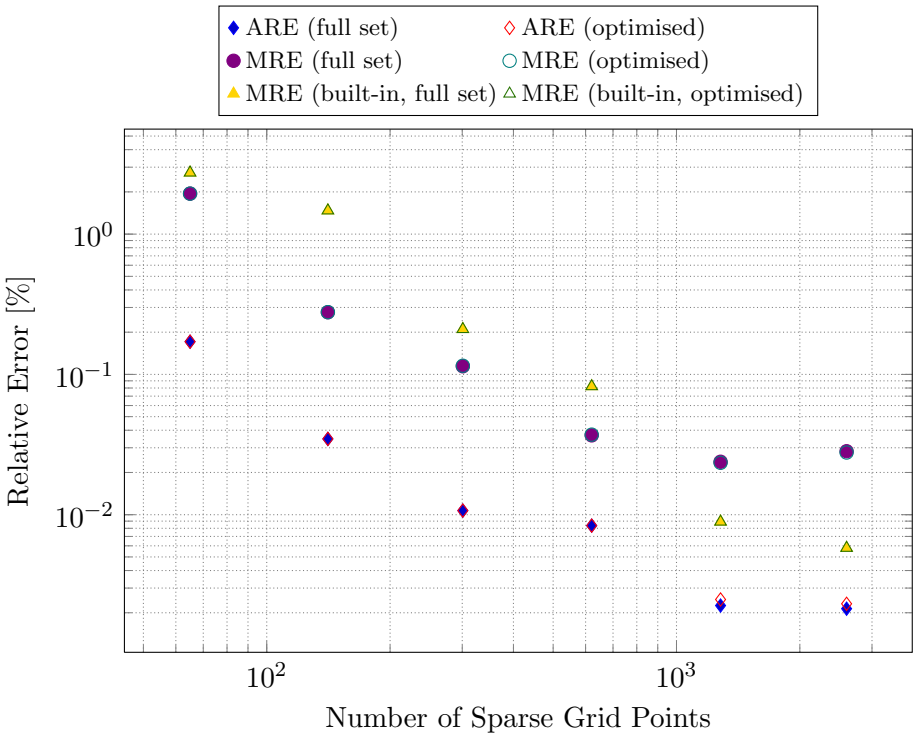


Figure 4.32: MTR assembly: interpolation errors on an anisotropic grid with $\boldsymbol{\alpha} = (1, 2, 2, 2)$

case the accuracy targets are not met, even at this extremely high level. What should also be kept in mind, though, is that the results of the most complex cross section dependencies at our disposal were presented. In the calculational path used by Necsa, the macroscopic cross sections are never transferred to the full core simulator. They were used as examples here because it is known that they present a more difficult problem to a representation method than microscopic cross sections. Results were also presented for the group six microscopic cross section of $^{238}_{92}$U in order to show that the method is indeed more successful for microscopic cross sections. Due to space and time limitations only results for this microscopic cross section are shown. Nevertheless, the results obtained for this microscopic cross section compare well with the accuracy of representing the thermal group cross sections for most of the important microscopic materials, including $^{235}_{92}$U absorption, fission and nu-fission and $^{135}_{54}$Xe absorption. The $^{240}_{94}$Pu cross sections for capture, fission and nu-fission in group four (epithermal) may be highlighted as the microscopic cross sections for which the representation methods performed the worst, with accuracy comparable to the group six macroscopic absorption cross section.

Thermal group cross sections were chosen as examples for two reasons. Firstly because the reactors under consideration are thermal reactors, and secondly because thermal cross section dependencies are more complex than those of fast group cross sections. This complexity was observed in this study, in the standard case PWR problem the interpolation of the group six macroscopic absorption cross section contained 1146 terms after optimisation and only achieved an average relative error of 3.8×10^{-2} %, while the group one (fast) macroscopic absorption cross section only required 62

Table 4.8: Nominal conditions and reduced intervals of variation for the PWR assembly

	nominal	min	max
Moderator temperature [K]	579.4	279.4	609.4
Fuel temperature [K]	951.4	291.4	1251.4
Boron concentration [ppm]	600	1	1200

terms for an average relative error of 3.6×10^{-3} % and a maximum relative error of 8.7×10^{-3} %. This accuracy was reached at grid level five, as opposed to grid level eight for the thermal cross section. Similarly, in the case of $^{238}_{92}$U absorption, the group six cross section needed 1128 terms for an average relative error of 8.0×10^{-3} %, while the group one cross section reached an average relative error of 7.7×10^{-4} % with only 63 terms, which had been found at level three already.

In terms of the energy group structure, cross sections become more challenging to represent if a larger number of groups is used. In only two groups, some of the more complex behaviour caused by phenomena such as resonances is averaged out, while the complexity is more pronounced if more groups are used. This consideration led us to use six groups for the PWR example instead of the more traditional two groups, to test the applicability of the method to more complex cross section models that may be used in future. This effect could be seen when the VVER example was compared to the PWR example. In both cases the macroscopic thermal absorption cross section was interpolated, but greater accuracy was achieved with fewer samples for the two-group VVER example than for the six-group PWR example.

As an extra challenge, the intervals of variation of the PWR problem were chosen to not only cover operational and start-up conditions, but also include values of the parameters that may only be reached during accident conditions. This was done intentionally to test whether the method is suitable for constructing cross section representations that will be used in transient analysis. Widening the state parameter intervals makes the cross section dependencies more complex. This was observed in a small study that was done into the effect of the state parameter intervals. When the intervals of fuel temperature, moderator temperature and Boron concentration were reduced to those given in Table 4.8, the error of approximation on an isotropic sparse grid of level seven was reduced by a factor of 2 for maximum relative error and a factor of 2.5 for average relative error of the group six macroscopic absorption cross section. These intervals would be sufficient for an operational library that is not intended for use in transient analysis. Interpolation was not done for these intervals in the PWR example.

Upon further investigation, another phenomenon that affects the accuracy of representation to such an extent that it needs to be accounted for, was uncovered. The combination of a finite step size in burnup and the method used to solve the Bateman equations in the transport code that was used, introduces an error in the calculated cross section values that is cumulative, and may become substantial over the lifetime of an assembly. There are two ways to account for this effect when

assessing the accuracy of the representation. One can ensure that all calculations use burnup steps that are small enough to eliminate this source of error. Alternatively, one can ensure that the burnup steps used to construct the representation are consistent with those used to calculate the independent test samples. Doing so would isolate the error of representation from the error inherent in the cross section calculation. In order to make the grids more consistent, one can adjust the size of the test grid to match the number of burnup steps in the sparse grid. One could also insert extra burnup points amongst either the sparse grid points or the test points, depending on which grid has more burnup points to start off with.

The isotropic sparse grids used in this work have $2^8 = 256$ nodes in any given dimension at level 8, and therefore requires 256 burnup points, with a maximum spacing of approximately 6×10^{-3} on the interval $[0, 1]$, or 360 MWd/tU in the physical burnup interval. A Sobol' grid is equidistant when projected onto any given dimension, which means that the 4096 quasi-random test points that were used have a spacing of approximately 2.4×10^{-4} on the interval $[0, 1]$ or 14.4 MWd/tU on the physical burnup interval. These grids are clearly not consistent, to get a test grid that is consistent with the sparse grid, 256 test points, which have a spacing of 4×10^{-3} on the unit interval, should be used. Such a small sample is not sufficiently statistically significant to give a reliable estimate of the maximum error, but may give a useful indication of the average error when the error introduced by the deficiencies of the transport calculation itself is removed.

Given the substantial effect that the disparity in step size may have, even after the Xe transient at BOL, new results were produced for the split burnup case of the PWR example. The calculation of the points used for constructing the representation was adapted by increasing the sparse grid level to 8 and manually adding 149 burnup points. In the current implementation, adding burnup points without off-bases to the pre-calculated grid points is done by hand, limiting the number that can be added this way. A set of test points with the same number of burnup steps as a level 8 sparse grid was produced for the same problem.

First the average relative error of the interpolation at sparse grid level 7 was calculated using the reduced test grid with 256 test points. The result was an ARE of 3.9×10^{-3} %, an order of magnitude smaller than the ARE calculated with 4096 test points. The accuracy of the interpolation on the adapted sparse grid was also measured using the original set of test points. The adaptations were found to have a significant effect; on a sparse grid of level 7 the maximum relative error was reduced from 8.3×10^{-2} % to 3.7×10^{-2} % and the average relative error from 5.8×10^{-2} % to 2.5×10^{-2} %. This is still larger than the required ARE, but may be improved further by refining the burnup mesh more. Even with the refinements discussed in this paragraph, the results show that the burnup effect still dominates the error.

The above discussion also explains the reason for the surprisingly good performance of the anisotropic sparse grid interpolation. The samples used to do the transport calculations came from a level 12 sparse grid, which has $2^{12} = 4096$ samples in the burnup dimension, the same as the number of test samples. These grids were therefore more consistent and a larger part of the burnup effect was

eliminated.

Chapter 5

Conclusions

A sparse grid quadrature module, based on both the Newton-Cotes and Clenshaw-Curtis quadrature rules, was developed. This module performs integration using isotropic or anisotropic sparse grids. It was verified on the Genz test functions and integrated into a cross section representation code that approximates cross section dependencies using quasi-regression. This quadrature module was used for all the approximations constructed as a part of this dissertation. Additionally, the functionality of the cross section representation code was expanded to allow the approximation of fission, nu-fission and transport cross sections. Finally the post-representation testing on quasi-random test samples was added to the cross section representation code.

A variety of sparse grid methods for representing homogenised few group cross section dependencies on state parameters were applied to three realistic test problems. Cross sections with Xenon build-up followed explicitly, as they are routinely calculated at Necsa, were used for the standard case. Several other approaches were used as variations: assuming that the Xenon number density was at equilibrium from the beginning of the assembly's life, splitting the burnup interval to create two libraries and using anisotropic instead of isotropic sparse grids. Only the case with equilibrium Xenon number density was used for the VVER example, whereas all the cases were used for the PWR and MTR examples. The PWR example was designed to be an especially difficult problem, with large intervals for the state parameters and more than the traditional two energy groups.

In the standard case, where isotropic sparse grids were used, the target accuracy of 0.1 % in maximum relative error and 0.01 % in average relative error in the macroscopic thermal absorption cross section was not achieved for the PWR example with either approximation or interpolation. For the MTR example, the target accuracy was not achieved using approximation, and required 2929 samples when using interpolation. In both examples an accurate representation of the thermal absorption cross section of the microscopic material $^{238}_{92}$U was successfully constructed with interpolation when using a high sparse grid level. Isotropic sparse grids are therefore not recommended if Xenon number density needs to be tracked explicitly.

Many problems may not need the initial Xenon transient to be modelled and the Xenon concentration may be set to its equilibrium value at the prevailing core conditions. In this case the accuracy of approximating and interpolating the macroscopic absorption cross section still fell short of the target accuracy for the PWR example. For the MTR example, approximation reached the target accuracy at a high grid level and interpolation did so at grid level 4, with 401 samples. In this case interpolation was used for the additional example of a VVER fuel pin, which reached the target accuracy for the macroscopic thermal absorption cross section at grid level 4, with 801 samples. Accurate representations were obtained for the MTR example and the VVER example with a reasonable number of samples using interpolation, therefore it is considered a viable representation method when Xenon number density does not need to be modelled explicitly at BOL.

To improve on the results of the standard case, the burnup interval was split to create a start-up library and an operational library. Good accuracy was achieved for the start-up library using either approximation or interpolation for both the MTR and PWR examples, but interpolation yielded a better rate of error decay for both examples. For the operational library, however, only interpolation of the MTR example was sufficiently accurate. The combined start-up plus operational representation for the MTR example required 538 samples to reach our accuracy limits when interpolation was used. Interpolation on an isotropic sparse grid is therefore a reasonable method for representing few group cross sections when the burnup interval is small, or for the MTR problem if the interval is split.

The standard case was repeated using anisotropic sparse grids with the anisotropy vector $\boldsymbol{\alpha} = (1, 2, 2, \ldots)$. Approximation was shown to not be an effective representation method using this anisotropy vector. Interpolation using this vector, on the other hand, produced an accurate representation of both the macroscopic and ${}^{238}_{92}\mathrm{U}$ absorption cross sections for the PWR example using 969 samples and for the MTR example using 301 samples. Interpolation on an anisotropic sparse grid with anisotropy vector $\boldsymbol{\alpha} = (1, 2, 2, \ldots)$ is the most efficient representation method that was studied and is recommended for all problems, especially highly challenging problems.

Consistently through all the cases that were investigated in this work, interpolation was more accurate than approximation by quasi-regression. This may be due to several factors. Firstly, the effective local support of interpolation with Lagrange basis functions may give it an advantage over approximation in terms of the balance between accuracy and the number of samples that are required. It is also possible that extra error was introduced into the approximation by using quasi-regression instead of standard regression. Clenshaw-Curtis quadrature may also not be the optimal quadrature to use when calculating the regression coefficients of a representation based on Legendre polynomials.

There are considerations other than accuracy to keep in mind when choosing a cross section representation method. In this study it was carefully ensured that every one of the transport calculations were fully converged to less than 1.0×10^{-6} % deviation in terms of the epithermal flux, the thermal flux and the fission source. There was therefore only a small numerical error associated

with each data point, though this did not address the error introduced by the finite burnup step size. Interpolation works well in this situation, as was demonstrated above. However if the transport solution is less accurate, approximation may be used to smooth out any "jitter" in the data.

On the whole then, the method was shown to have enough flexibility to handle difficult problems in a consistent way. The built-in error measures are not always accurate, but they are conservative for the most part.

5.1 Future Work

Several questions were not addressed in this work and will be investigated in the near future. The performance of the cross section representation method on increasingly difficult problems, such as MOX fuel and fuel that contains burnable absorbers, will be established. A full review of results for all microscopic cross sections will be carried out. The improvement in accuracy that may result from replacing the quasi-regression with standard regression will also be investigated.

In the longer term, the effect of using this cross section representation method to produce cross sections for use in the full core simulator will be investigated. This will require some changes to the simulator, and the creation of a library that contains all the extra information that the simulator requires, such as decay chains. It is also intended to use Xenon number density as a state parameter, but in order to do this changes need to be made to the assembly code and the core simulator.

The final recommendation of this dissertation is to develop the test code that performs anisotropic sparse grid interpolation into a production-ready program. This program should then replace the cross section representation code that is currently used for day-to-day reactor calculations at Necsa. The capabilities offered by ANOVA should also be available in the production-ready system when need. An option that is possible but that was not discussed in this dissertation is ANOVA analysis using interpolation rather than approximation. It is thus recommended that approximation on isotropic sparse grids be incorporated into the new cross section representation system, or that ANOVA analysis using interpolation be developed.

Publications that resulted from the work done for this thesis

1. R. H. Prinsloo and P. M. Bokov and G. Stander and D. Botes and W. Joubert. An automated cross section parametrization method for MTR application. *International Conference on Mathematics, Computational Methods & Reactor Physics*, Saratoga Springs, New York, 3 – 7 May 2009.

2. D. Botes and P. M. Bokov. Hierarchical, multilinear representation of few-group cross sections on sparse grids. *M&C 2011 International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Rio de Janeiro, Brazil, 8 – 12 May 2011.

3. P. M. Bokov and D. Botes and V. G. Zimin. Pseudospectral Chebyshev representation of few-group cross sections on sparse grids. *PHYSOR 2012 Advances in Reactor Physics Linking Research, Industry, and Education*, Knoxville, Tennessee, 15 – 20 April 2012.

Bibliography

An, J. and A. Owen (2001). "Quasi-regression". In: *Journal of Complexity*, **17**(4), pp. 588–607.

Barthelmann, V., E. Novak, and K. Ritter (2000). "High dimensional polynomial interpolation on sparse grids". In: *Advances in Computational Mathematics*, **12**(4), pp. 273–288.

Bellman, R. E. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton University Press.

Berrut, J. P. and L. N. Trefethen (2004). "Barycentric Lagrange interpolation". In: *SIAM Review*, **46**(3), pp. 501–517.

Boas, M. L. (2006). *Mathematical Methods in the Physical Sciences*. 3rd ed. John Wiley & Sons, Inc.

Bokov, P. M., D. Botes, and V. G. Zimin (Apr. 15–20, 2012). "Pseudospectral Chebyshev representation of few-group cross sections on sparse grids". In: *PHYSOR 2012 Advances in Reactor Physics Linking Research, Industry, and Education*. Knoxville, Tennessee.

Bokov, P. M. and R. Prinsloo (Apr. 15–19, 2007). "Cross-section parameterization using quasi-regression approach". In: *Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications (M&C + SNA 2007)*. Monterey, California.

Bokov, P. (2009). "Automated few-group cross-section parameterization based on quasi-regression". In: *Annals of Nuclear Energy*, **36**(8), pp. 1215–1223.

Botes, D. and P. M. Bokov (May 8–12, 2011). "Hierarchical, multilinear representation of few-group cross sections on sparse grids". In: *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011)*. Rio de Janeiro, RJ, Brazil.

Boyack, B. E. et al. (1990). "Quantifying reactor safety margins part 1: An overview of the code scaling, applicability, and uncertainty evaluation methodology". In: *Nuclear Engineering and Design*, **119**(1), pp. 1–15.

Boyd, J. P. (2000). *Chebyshev and Fourier Spectral Methods*. DOVER Publications, Inc.

Brown, F. B. et al. (2002). "MCNP version 5". In: *Transactions of the American Nuclear Society*, **87**, pp. 273–276.

Burden, R. L. and J. D. Faires (2001). *Numerical Analysis*. 7th ed. Brooks/Cole.

Burkardt, J. (2009). *Source Codes in Fortran90 [Online]*. URL: `http://people.sc.fsu.edu/~jburkardt/f_src/f_src.html`.

Burkardt, J., M. D. Gunzburger, and C. Webster (2007). "Reduced order modeling of some nonlinear stochastic partial differential equations". In: *International Journal of Numerical Analysis and Modeling*, **4**(3–4), pp. 368–391.

Chadwick, M. et al. (2006). "ENDF/B-VII.0: Next generation evaluated nuclear data library for nuclear science and technology". In: *Nuclear Data Sheets*, **107**(12). Evaluated Nuclear Data File ENDF/B-VII.0, pp. 2931–3060.

Chang, G. S. (Apr. 17–21, 2005). "MCWO–Linking MCNP and ORIGEN2 for fuel burnup analysis". In: *Proceedings of The Monte Carlo Method: Versatility Unbounded In A Dynamic Computing World*. American Nuclear Society. Chattanooga, Tennessee.

Chibani, O. and X. A. Li (2002). "Monte Carlo dose calculations in homogeneous media and at interfaces: A comparison between GEPTS, EGSnrc, MCNP, and measurements". In: *Medical Physics*, **29**(5), pp. 835–848.

Churchhouse, R. F. et al. (1981). *Numerical Methods*. Ed. by R. F. Churchhouse. Vol. 3. Handbook of Applicable Mathematics. John Wiley & Sons.

Corput, J. G. van der (1935). "Verteilungsfunktionen I,II". In: *Nederl. Akad. Wetensch. Proc.* **38**, pp. 813–820, 1058–1066.

Davidenko, V. D. and V. F. Tsibulsky (Oct. 5–8, 1998). "Detailed calculation of neutron spectrum in cell of a nuclear reactor". In: *International Conference on the Physics of Nuclear Science and Technology*. New York, USA, pp. 1755–1760.

DeVore, R. A. and G. G. Lorentz (1993). *Constructive Approximation*. A Series of Comprehensive Studies in Mathematics. Springer-Verlag.

Diop, C. M. et al. (Mar. 14–17, 2007). "TRIPOLI-4: A 3D continuous-energy Monte Carlo transport code". In: *PHYTRA1: First International Conference on Physics and Technology of Reactors and Applications*. Marrakech, Morocco.

Downar, T., C. H. Lee, and G. Jiang (May 7–12, 2000). "An assessment of advanced nodal methods for MOX fuel analysis in light water reactors". In: *Proceedings of the PHYSOR 2000 ANS International Topical Meeting on Advances in Reactor Physics and Mathematics and Computation into the Next Millennium*. Pittsburgh, Pennsylvania.

Duderstadt, J. J. and L. J. Hamilton (1976). *Nuclear Reactor Analysis*. John Wiley & Sons New York.

Dufek, J. (2011a). "Building the nodal nuclear data dependences in a many-dimensional state-variable space". In: *Annals of Nuclear Energy*, **38**, pp. 1569–1577.

— (May 8–12, 2011b). "Complex models of nodal nuclear data". In: *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011)*. Rio de Janeiro,Brazil.

Everaars, K. and B. Koren (1997). "Using coordination to parallelize sparse-grid methods for 3D CFD problems". In: *Parallel Computing*, **24**(7), pp. 1081 –1106.

Fujita, T. et al. (May 9–14, 2010). "Investigation on macroscopic cross section model for BWR pin-by-pin core analysis". In: *PHYSOR 2010–Advances in Reactor Physics to Power the Nuclear Renaissance*. Pittsburgh, Pennsylvania, USA.

Genz, A. C. (1984). "Tools, Methods and Languages for Scientific and Engineering Computation". In: *Testing Multiple Integration Software*. Ed. by B. Ford, J. C. Rault, and F. Thomasset. North-Holland, pp. 81–94.

Gerstner, T. (2007). "Sparse Grid Quadrature Methods for Computational Finance". PhD thesis. Rheinischen Friedrich-Wilhelms-Universität Bonn.

Gerstner, T. and M. Griebel (1998). "Numerical integration using sparse grids". In: *Numerical Algorithms*, **18**(3), pp. 209–232.

Geyer, C. (1994). "On the convergence of Monte Carlo maximum likelihood calculations". In: *Journal of the Royal Statistical Society Series B*, **56**, pp. 261–274.

Griebel, M. (2006). *Sparse Grids and Related Approximation Schemes for Higher Dimensional Problems*. Sonderforschungsbereich 611, Singuläre Phänomene und Skalierung in Mathematischen Modellen. SFB 611.

Griebel, M. and J. Hamaekers (2007). "Sparse grids for the Schr odinger equation". In: *ESAIM: Mathematical Modelling and Numerical Analysis*, **41**, pp. 215–247.

Guimaraes, P. F. and E. Müller (May 3–7, 2009). "Parameterization of two-group nodal cross section data for POLKA-T BWR transient applications". In: *International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009)*. Saratoga Springs, New York.

Hébert, A. (2009). *Applied Reactor Physics*. Presses Internationale Polytechnique.

Klimke, A. (2006). *Efficient construction of hierarchical polynomial sparse grid interpolants using the fast discrete cosine transform*. Tech. rep. Technical Report IANS Preprint 2006/007, Universität Stuttgart, 2006. 14.

Knott, D. and A. Yamamoto (2010). "Reactor Design". In: ed. by D. G. Cacuci. Vol. 2. Handbook of Nuclear Engineering. Springer. Chap. Lattice Physics Computations, pp. 913–1241.

Kosaka, S. and E. Saji (2000). "Transport theory calculation for a heterogeneous multi-assembly problem by characteristics method with direct neutron path linking technique". In: *Journal of Nuclear Science and Technology*, **37**(12), pp. 1015–1023.

Lemieux, C. and A. B. Owen (2002). "Quasi-regression and the relative importance of the ANOVA components of a function". In: *Monte Carlo and Quasi-Monte Carlo Methods 2000: Proceedings of a Conference Held at Hong Kong Baptist University*. Ed. by K.-T. Fang, F. Hickernell, and H. Niederreiter. Hong Kong SAR, China: Springer-Verlag, pp. 331–344.

Leppänen, J. (2007). "Development of a New Monte Carlo Reactor Physics Code". PhD thesis. Helsinki University of Technology.

Leszczynski, F., D. L. Aldama, and A. Trkov (2007). *WIMS-D library update: Final report of a coordinated research project*. Tech. rep. IAEA.

Mathé, P. and E. Novak (2007). "Simple Monte Carlo and the Metropolis algorithm". In: *Journal of Complexity*, **23**(4), pp. 673–696.

Metropolis, N. and S. Ulam (1949). "The Monte Carlo method". In: *Journal of the American Statistical Association*, **44**(247), pp. 335–341.

Müller, E., L. Mayhue, and B. Zhang (Sept. 10–13, 2007). "Reactor physics methods development at Westinghouse". In: *International Conference Nuclear Energy for New Europe 2007*. Portorož, Slovenia.

Müller, E. Z. et al. (1994). "Development of a core follow calculational system for research reactors". In: *Transactions of the American Nuclear Society*, **70**, pp. 1047–1051.

Müller, E. Z. (July 1987). *A review of some cross section generation methodologies for nodeal analysis of pressurised water reactors*. Tech. rep. RTE01–2/2–002. South African Nuclear Energy Corporation.

Nahm, T. (2005). "Error estimation and index refinement for dimension-adaptive sparse grid quadrature with applications to the computation of path integrals". MA thesis. University of Bonn.

Nobile, F., R. Tempone, and C. G. Webster (2008a). "A sparse grid stochastic collocation method for partial differential equations with random input data". In: *SIAM Journal on Numerical Analysis*, **46**(5), pp. 2309–2345.

— (2008b). "An anisotropic sparse grid stochastic collocation method for elliptic partial differential equations with random input data". In: *SIAM Journal on Numerical Analysis*, **46**(5), pp. 2411–2442.

Novak, E. and K. Ritter (1996). "High dimensional integration of smooth functions over cubes". In: *Numerische Mathematik*, **75**(1), pp. 79–97.

Prinsloo, R. H. et al. (May 3–7, 2009). "An automated cross section parametrization method for MTR application". In: *International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009)*. Saratoga Springs, New York.

Rabitz, H. et al. (1999). "Efficient input-output model representations". In: *Computer Physics Communications*, **117**, pp. 11–20.

Reisinger, C. and G. Wittum (2007). "Efficient hierarchical approximation of high-dimensional option pricing problems". In: *SIAM Journal on Scientific Computing*, **29**(1), pp. 440–458.

Reitsma, F. and W. R. Joubert (Mar. 28–30, 1999). "A calculational system to aid economical use of MTRs". In: *International Conference on Research Reactor Fuel Management (RRFM'99)*. Brugges, Belgium.

Sato, D. et al. (May 9–14, 2010). "A new robust cross section representation methodology for PWR core simulator". In: *PHYSOR 2010–Advances in Reactor Physics to Power the Nuclear Renaissance*. Pittsburgh, Pennsylvania.

Smoljak, S. A. (1963). "Quadrature and interpolation formulas for tensor products of certain classes of functions". In: *Doklady Mathematics*, **4**, pp. 240–243.

Sobol', I. M. (1967). "On the distribution of points in a cube and the approximate evaluation of integrals". In: *USSR Computational Mathematics and Mathematical Physics*, **7**(4), pp. 86 –112.

— (1993). "Sensitivity estimates for nonlinear mathematical models". In: *Mathematical Modeling and Computational Experiment*, **1**, pp. 407–414.

— (1998). "On quasi-Monte Carlo integrations". In: *Mathematics and Computers in Simulation*, **47**(2–5), pp. 103–112.

— (2003). "Theorems and examples on high dimensional model representation". In: *Reliability Engineering and System Safety*, **79**(2), pp. 187–193.

Spanier, J. (2010). "Monte Carlo Methods". In: *Nuclear Computational Science: A Century in Review*. Ed. by Y. Azmy and E. Sartori. Springer, pp. 117–166.

Stacey, W. M. (2001). *Nuclear Reactor Physics*. John Wiley & Sons New York.

Trkov, A. (Mar. 13–Apr. 14, 2000). "From basic nuclear data to applications". In: *Workshop on Nuclear Data and Nuclear Reactors: Physics, Design and Safety*. Trieste, Italy, pp. 321 –346.

Turski, R. B. et al. (Oct. 6–10, 1997). "Macroscopic cross section generation and application for coupled spatial kinetics and thermal hydraulics analysis with SAS-DIF3DK". In: *Joint International Conference on Mathematical Methods and Supercomputing for Nuclear Applications, Saratoga '97*. Saratoga Springs, New York.

Wasilkowski, G. W. and H. Wozniakowski (1995). "Explicit cost bounds of algorithms for multivariate tensor product problems". In: *Journal of Complexity*, **11**(1), pp. 1–56.

Watson, J. K. and K. N. Ivanov (2002). "Improved cross-section modeling methodology for coupled three-dimensional transient simulations". In: *Annals of Nuclear Energy*, **29**(8), pp. 937–966.

Zimin, V. G. and A. A. Semenov (2005). "Building neutron cross-section dependencies for few-group reactor calculations using step-wise regression". In: *Annals of Nuclear Energy*, **32**(1), pp. 119–136.

Živanović, R. and P. M. Bokov (2010). "Cross-section parameterization of the pebble bed modular reactor using the dimension-wise expansion model". In: *Annals of Nuclear Energy*, **37**(12), pp. 1763–1773.