

# A Modal Expansion Equilibrium Cycle Perturbation Method for Optimizing High Burnup Fast Reactors

by

Nicholas W. Touran

A dissertation submitted in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
(Nuclear Engineering and Radiological Sciences)  
in The University of Michigan  
2012

Doctoral Committee:

Professor John C. Lee, Chair  
Professor Thomas J. Downar  
Professor William R. Martin  
Associate Professor Divakar Viswanath

© Nicholas W. Touran 2012  
All Rights Reserved

For Dad.

## ACKNOWLEDGEMENTS

I am first and foremost thankful for the opportunity to perform this research, which was gracefully provided by my parents, along with their early guidance, inspiration, and support, and by everyone throughout humanity's complex history who enabled the overall societal order and environment conducive to my pursuit of science and engineering. A very special thanks to Professor Lee for his wise advice and kindness throughout the years, for his training and inspiration that led me into my current interests and skills, for his strong role in motivating and developing this research, and for his infinite patience. Thanks to those who kindly assisted in keeping me on the path, including Lowell Wood for continually badgering me and neglecting to invite me to interesting meetings. Also in this category are Jon McWhirter, Robert Petroski, Roger Reynolds, Dave McAlees, Caroline Bisk, and many others (you know who you are). Thanks to Jesse Cheatham for shouldering other responsibilities I was involved with while finalizing this work. Infinite thanks to John Gilleland, Pavel Hejzlar, and the rest of the TerraPower team for providing the means and flexibility to perform the bulk of this work, as well as for doing pioneering work in TWR core design and model development. Thanks to Jeff Davis for my early training in fast reactor tools, to Dan Wessol for introducing me to Python, and to Chris Rycroft

for introducing me to and tutoring me in most of the other tools used to perform this research. And, of course, a huge thank you to all my classmates, professors, friends, and family who helped in all regards along the way. Lastly, a big thanks to the nameless peer advisor who, almost a decade ago, informed me of the wonderful field of nuclear engineering.

# TABLE OF CONTENTS

<b>DEDICATION</b> . . . . .	ii
<b>ACKNOWLEDGEMENTS</b> . . . . .	iii
<b>LIST OF FIGURES</b> . . . . .	ix
<b>LIST OF TABLES</b> . . . . .	xii
<b>LIST OF ABBREVIATIONS</b> . . . . .	xiv
<b>ABSTRACT</b> . . . . .	xvi
<b>CHAPTER</b>	
<b>I. Introduction</b> . . . . .	1
1.1 Primary Considerations in Reactor Design . . . . .	3
1.2 Motivations for High-Burnup Fast Reactors . . . . .	6
1.3 The Evolution of Perturbation Methods for Reactor Design . . . . .	9
1.3.1 Variational Methods . . . . .	9
1.3.2 Explicit High-Order Perturbation Theory . . . . .	10
1.3.3 Time-Dependent Perturbation Theory . . . . .	12
1.4 Nuclear Reactor Design Optimization . . . . .	14
1.5 The Objective and Organization of this Thesis . . . . .	16
<b>II. An Enhanced Equilibrium Cycle Method</b> . . . . .	20
2.1 The Traditional Equilibrium Cycle Methodology . . . . .	21

2.1.1	Mathematical Representation of the Reactor . . . . .	23
2.1.2	Formulation of the Equilibrium Cycle . . . . .	25
2.1.3	Solution of the Equilibrium Cycle . . . . .	30
2.2	Improvements to the Equilibrium Cycle Methodology . . . . .	36
2.2.1	Description of the ARMI . . . . .	37
2.2.2	Fuel Performance . . . . .	42
2.2.3	The Burnup-Dependent Cross Section Model . . . . .	43
2.2.4	Implementation of Enhanced Equilibrium Cycle . . . . .	45
2.3	Equilibrium Calculations with EEC . . . . .	49
2.3.1	Description of the Reference Core . . . . .	49
2.3.2	Investigation of Burn Matrix Averaging . . . . .	51
2.3.3	The Effects of Burnup Dependent Cross Sections on the Equilibrium Search . . . . .	57
2.3.4	The Effects of Coupled Fuel Performance on the Equilibrium Search . . . . .	57
2.3.5	The High Burnup Equilibrium Instability . . . . .	59
<b>III. Modal Expansion Perturbation Theory . . . . .</b>		<b>64</b>
3.1	Derivation of Modal Expansion Perturbation Theory . . . . .	64
3.1.1	The Finite-Difference Solution of the Steady-State Neutron Diffusion Equation . . . . .	65
3.1.2	Determining an Algebraic Expression of Perturbed Flux for Arbitrary Perturbations . . . . .	75
3.1.3	Discussion of the Eigenvalue Expansion . . . . .	80
3.1.4	The Role of the Perturbed Eigenvalue . . . . .	81
3.1.5	MEPT In the Condensed Fission Source Problem . . . . .	83
3.1.6	Computational Efficiency of MEPT . . . . .	85
3.1.7	Parameters Available Through MEPT . . . . .	87
3.1.8	Arnoldi's Method for Calculating the Modal Har- monics . . . . .	88
3.1.9	Relation to Other Methods . . . . .	91
3.2	Implementation of MEPT within the ARMI Framework . . . . .	94
3.2.1	Development of the New Diffusion Code DIFNT . . . . .	95
3.2.2	Validation of DIFNT Against DIF3D . . . . .	96
3.2.3	Using DIFNT to Calculate Modal Perturbation Co- efficients . . . . .	97
3.3	Calculating Perturbed Power and Flux Distributions with MEPT . . . . .	99

3.3.1	Performance of MEPT with Perturbations in an Assembly Cluster . . . . .	99
3.3.2	Performance of MEPT with Modeling Perturbations to a Single Assembly . . . . .	106
3.3.3	Performance of MEPT with Modeling Perturbations to a Single Block of a Single Assembly . . . . .	108
<b>IV. Equilibrium Cycle Perturbation Theory . . . . .</b>		<b>110</b>
4.1	A First-Order Depletion Approximation . . . . .	110
4.1.1	Derivation of the First-Order Depletion Approximation . . . . .	112
4.1.2	Performance of the FODA . . . . .	113
4.2	Applying Modal Expansion Perturbation Theory to the Equilibrium Cycle . . . . .	116
4.2.1	Description of ECPT . . . . .	116
4.2.2	The Region-density iteration with FODA . . . . .	117
4.2.3	Implementation of ECPT within ARMI . . . . .	119
4.3	Numerical Demonstration of ECPT . . . . .	121
4.3.1	Perturbed flux and direct depletion . . . . .	122
4.3.2	Perturbed flux and perturbed depletion . . . . .	124
<b>V. Multi-Objective Nuclear Reactor Design Optimization . . . . .</b>		<b>127</b>
5.1	Creating a Surrogate Model of the Reactor . . . . .	127
5.1.1	Nonlinear Regressions with the Alternating Conditional Expectation Algorithm . . . . .	128
5.1.2	Training the surrogate model with ARMI . . . . .	131
5.1.3	Sample Surrogate of a High-Burnup Fast Reactor . . . . .	134
5.2	Multi-Objective Optimization with Physical Programming . . . . .	139
5.2.1	Multi-Objective Optimization Background . . . . .	139
5.2.2	Introduction to Physical Programming . . . . .	142
5.2.3	Mathematical Construction of the Class Functions . . . . .	146
5.2.4	Minimizing The Aggregate Objective Function . . . . .	152
5.2.5	Sample Optimization of a Simple Nuclear Reactor Model . . . . .	154
5.3	Optimization of a High Burnup Fast Reactor . . . . .	157

5.3.1	Comparison between direct and perturbation-based evaluations . . . . .	159
5.3.2	Demonstration of changing preferences . . . . .	163
<b>VI.</b>	<b>Summary and Conclusions . . . . .</b>	<b>165</b>
6.1	Potential Improvements . . . . .	167
6.2	Alternative Applications . . . . .	170
<b>REFERENCES</b>	<b>. . . . .</b>	<b>172</b>

## LIST OF FIGURES

### Figure

2.1	A simple visualization of a fuel management chain . . . . .	27
2.2	Flowchart of the REBUS-3 equilibrium cycle. . . . .	31
2.3	Flowchart of the REBUS cyclic mode iteration. . . . .	32
2.4	The Enhanced Equilibrium Cycle flow chart. . . . .	48
2.5	Convergence of a direct equilibrium cycle calculation in terms of $k_{eff}$ . The value oscillates around the final value and was randomly very close to it around iteration 55, but this should not be confused for a converged value. Recall that the convergence criteria in this case is driven by the region number densities. The eigenvalue convergence for each cyclic iteration was such that all digits are relevant. . . . .	53
2.6	Convergence of a equilibrium-mode calculation in terms of $k_{eff}$ . . .	53
2.7	Convergence in terms of peak number density error between cyclic iterations in a direct equilibrium calculation. The sharp drop repre- sents the iteration where all the fresh fuel had discharged from the core. . . . .	54
2.8	Convergence in terms of peak number density error between cyclic iterations in a fast equilibrium-mode calculation. . . . .	54
2.9	Effects of burnup dependent cross sections on BOEC $k_{eff}$ . . . . .	58
2.10	Effects of fuel performance feedback on BOEC $k_{eff}$ . . . . .	60
2.11	Power of a single assembly in an equilibrium case demonstrating the high burnup equilibrium instability. . . . .	62
2.12	Power of a single assembly in an explicitly shuffled equilibrium case demonstrating the stability of the explicit method. . . . .	62
3.1	The $A_g$ matrix of a 3-D finite-difference mesh of a small 1/3-symmetric triangular sample problem with three z-direction mesh points. . . . .	69
3.2	The $Q$ matrix structure from a small sample problem. . . . .	73

3.3	The 33-group flux spectrum computed by DIFNT compared to that of DIF3D. . . . .	98
3.4	The first 20 real flux harmonics of a bare reactor sample case as calculated by DIFNT. The harmonics are summed over all 33 energy groups. . . . .	100
3.5	The 90 largest real eigenvalues of a bare sample problem as calculated by DIFNT. . . . .	100
3.6	The uranium mass enrichment distribution (in %) of the perturbed cluster of assemblies. . . . .	101
3.7	The deviation in flux (in %) between the reference state and the directly-calculated perturbed state for a localized 20% increase in enrichment in a cluster of 7 assemblies. . . . .	102
3.8	The deviation in flux (in %) between the reference state and the MEPT-calculated perturbed state for a localized 20% increase in enrichment in a cluster of assemblies. The color scale is identical to the uncorrected comparison above. . . . .	102
3.9	Expansion coefficients for a perturbation on an assembly cluster . . .	103
3.10	Performance of MEPT for an assembly cluster . . . . .	104
3.11	Core-average and point fractional flux deviations of 1% enrichment perturbation on an offset cluster of assemblies. . . . .	105
3.12	Maximum point flux deviations from reference of the 1st- and 2nd-order MEPT formulations with $\lambda'_0$ at various points between $\lambda_0$ and the directly-calculated perturbed eigenvalue. . . . .	106
3.13	Expansion coefficients for a single assembly perturbation . . . . .	107
3.14	Performance of MEPT for a single assembly . . . . .	107
3.15	The first 400 expansion coefficients for a 20% increase in uranium enrichment in the bottom block of a single assembly in ring position 5. . . . .	108
3.16	Performance of MEPT for a single block . . . . .	109
4.1	The performance of the FODA for several important nuclides, plus the effect of adding the $\delta A$ term. . . . .	115
4.2	Equilibrium Cycle Perturbation Theory flowchart. . . . .	118
4.3	The standard region-density iteration. . . . .	120
4.4	The FODA-based region-density iteration. . . . .	120
4.5	Expansion coefficients for the 68% smear density ECPT case with direct depletion. . . . .	125
5.1	Sample regression quality for a critical case with a 100 pcm convergence criterion. . . . .	133

5.2	Regression quality for a critical case with a 1000 pcm convergence criterion. . . . .	133
5.3	The ACE transforms for the maximum discharge burnup. The first three plots represent transforms of the independent variables, and the fourth shows the transform of the dependent variable. . . . .	138
5.4	The ACE transforms for the equilibrium fuel cycle cost rate. . . . .	140
5.5	Flow charts showing the advantage of the physical programming approach over traditional weighting methods. . . . .	144
5.6	General shapes of the class functions . . . . .	145
5.7	Physically-meaningful class function definitions . . . . .	147
5.8	An example Class 1 “smaller is better” function. . . . .	153
5.9	An example Class 2 “larger is better” function. Note that the class values at each point are identical to those in Figure 5.8. . . . .	153
5.10	The core map for the high burnup fast reactor optimization case. . .	158
5.11	The 40th adjoint reference harmonic. The slight asymmetry in the 1/6 core is expected due to the asymmetric fuel management path. The localized nature of this high harmonic is evident. The adjoint harmonics are not normalized. . . . .	159
5.12	The ACE regression results for the high burnup fast reactor case. .	160

## LIST OF TABLES

### Table

2.1	Major Features of ARMI. . . . .	39
2.2	Reference TWR core design parameters. . . . .	50
2.3	Comparison of the $k_{eff}$ and discharge burnup of the direct and equilibrium mode cases. . . . .	51
2.4	The effects of burn matrix averaging and burn step length. The reference case is the Direct2 case, with $k_{eff} = 1.010717$ . . . . .	56
2.5	The effects of burnup dependent microscopic cross sections on the equilibrium cycle methodology. . . . .	56
2.6	The effects of coupled fuel performance, including axial expansion, fission gas removal, and bond squeeze-out on the equilibrium cycle methodology. $\Delta k_{eff}$ is compared to the highest-fidelity case treating all effects. . . . .	59
3.1	DIFNT results compared with DIF3D reference . . . . .	97
4.1	Comparison of ECPT calculations with direct depletion to EEC reference with smear density perturbations. . . . .	123
4.2	Comparison of ECPT calculations of the discharge burnup in %FIMA with FODA depletion to EEC cases with smear density perturbations. . . . .	126
5.1	The 125-point test matrix for the sample surrogate model. . . . .	135
5.2	Standard deviation in the fractional error between measured points and their surrogate values for all points and for a predictive 90% model. The average total error has been accounted for in the $\theta(Y)$ transform. The standard deviations for the predictive case are lower because only 10% of the measured points were compared vs. 100% in the first column. . . . .	136
5.3	The class parameters defined in Physical Programming lexicon. . . . .	145
5.4	Physical Programming region boundaries for sample problem . . . . .	156

5.5	Optimization results from various optimization algorithms . . . . .	156
5.6	The 125-point test matrix for the sample TWR optimization model. All points were executed with ECPT using a total of 25 reference points. . . . .	157
5.7	Physical Programming region boundaries for the high-burnup fast reactor optimization . . . . .	162
5.8	High-burnup reactor optimization results from direct and ECPT calculations. Results from the L-BFGS-B optimization engine are shown for both cases, although all others return nearly identical re- sults. . . . .	162
5.9	Alternate Physical Programming region boundaries for the high- burnup fast reactor optimization. . . . .	164
5.10	High-burnup reactor optimization results from the alternate prefer- ence set. . . . .	164

## LIST OF ABBREVIATIONS

<b>BOL</b>	beginning of life
<b>BOC</b>	beginning of cycle
<b>BOEC</b>	beginning of equilibrium cycle
<b>EOC</b>	end of cycle
<b>EOEC</b>	end of equilibrium cycle
<b>EOL</b>	end of life
<b>EEC</b>	Enhanced Equilibrium Cycle
<b>MEPT</b>	Modal Expansion Perturbation Theory
<b>ECPT</b>	Equilibrium Cycle Perturbation Theory
<b>ARMI</b>	Advanced Reactor Modeling Interface
<b>TWR</b>	Traveling Wave Reactor
<b>LWR</b>	light-water reactor
<b>SFR</b>	sodium-cooled fast reactor
<b>DPT</b>	Depletion Perturbation Theory
<b>GPT</b>	Generalized Perturbation Theory
<b>GUI</b>	graphical user interface

**T/H** thermal hydraulics  
**CRBR** Clinch River Breeder Reactor  
**FCMI** fuel-cladding mechanical interaction  
**FCCI** fuel-cladding chemical interaction  
**FIMA** fissions per initial heavy metal atom  
**LFP** lumped fission product  
**CR** control rod  
**IFR** Integral Fast Reactor  
**MDO** multidisciplinary design optimization  
**ACE** Alternating Conditional Expectation  
**PP** Physical Programming  
**MOO** multi-objective optimization  
**FODA** First-Order Depletion Approximation  
**IRAM** implicitly-restarted Arnoldi method  
**SOR** successive over-relaxation  
**GMRES** generalized minimal residual  
**CG** conjugate gradients  
**FMSR** Fast Mixed-Spectrum Reactor  
**DPA** displacements per atom  
**SM** Standard Method  
**LMFBR** liquid-metal cooled fast breeder reactor  
**EBR-II** Experimental Breeder Reactor Unit 2  
**FFTF** Fast Flux Test Facility

## ABSTRACT

This dissertation develops a simulation tool capable of optimizing advanced nuclear reactors considering the multiobjective nature of their design. An Enhanced Equilibrium Cycle (EEC) method based on the classic equilibrium method is developed to evaluate the response of the equilibrium cycle to changes in the core design. Advances are made in the consideration of burnup-dependent cross sections and dynamic fuel performance (fission gas release, fuel growth, and bond squeeze-out) to allow accuracy in high-burnup reactors such as the Traveling Wave Reactor. EEC is accelerated for design changes near a reference state through a new modal expansion perturbation method that expands arbitrary flux perturbations on a basis of  $\lambda$ -eigenmodes. A code is developed to solve the 3-D, multigroup diffusion equation with an Arnoldi-based solver that determines hundreds of the reference flux harmonics and later uses these harmonics to determine expansion coefficients required to approximate the perturbed flux. The harmonics are only required for the reference state, and many substantial and localized perturbations from this state are shown to be well-approximated with efficient expressions after the reference calculation is performed. The modal expansion method is coupled to EEC to produce the later-in-time response of each design perturbation.

Because the code determines the perturbed flux explicitly, a wide variety of core performance metrics may be monitored by working within a recently-developed data management system called the ARMI. Through ARMI, the response of each design perturbation may be evaluated not only for the flux and reactivity, but also for reactivity coefficients, thermal hydraulics parameters, economics, and transient performance.

Considering the parameters available, an automated optimization framework is designed and implemented. A non-parametric surrogate model using the Alternating Conditional Expectation (ACE) algorithm is trained with many design perturbations and then transformed through the Physical Programming (PP) paradigm to build an aggregate objective function without iteratively determining weights. Finally, the design is optimized with standard gradient-based methods. Through the power of ACE and the transparency of PP, the optimization system allows users to locate designs that best suit their multiobjective preferences with ease.

## CHAPTER I

### Introduction

Society will continue to rely on technological innovation to improve the worldwide standard of living as the population continues to increase. Many technologies require a great deal of energy to succeed, but concerns over our energy supply are on the rise. To perpetuate society's capability to adapt, we must stay agile in our ability to react to environmental and economic pressures on energy supplies.

Energy is generally available on Earth through five known mechanisms: continuous radiative energy from the sun (solar, wind, hydro, bio, wave), historical radiative energy from the sun (coal, oil, natural gas), residual heat from the formation of the solar system (geothermal), energy stored in the Earth-Moon rotational system (tidal), and energy stored in the nuclei of small atoms (fusion) and in uranium- or thorium-bearing minerals (fission). Of these, the first is the largest in magnitude, with the sun continually bathing us in petawatts of energy. This extreme amount of energy, however, is spread over the entire surface of the Earth, and is intermittent, so far preventing it from being used on a large scale. Meanwhile, concerns over sup-

ply and environmental consequences of both fossil fuels and traditional nuclear fuel leave them immersed in plenty of debate. Fusion still holds promise, but we have yet to capture it in a practical way. The others have fringe uses but cannot yet power society.

Advanced nuclear energy systems have been built and operated that had the potential to enhance nuclear fission into a clearly superior energy source for the future. So far, none of these systems have been adopted for a variety of reasons, including economic and political concerns. Regardless, we know for certain that these reactors and fuel cycles exist and work. They can provide millenia of world-scale energy while producing very little (albeit highly-toxic) waste. They have passive safety features and built-in proliferation resistance. With development, they appear to be able to compete economically with mainstream energy sources. It behooves mankind to keep actively pursuing advanced nuclear power from all angles, including fuel and material development, reactor design, fuel cycle safeguards, safety, and capital cost reductions.

Reactor designers face a large set of coolants, structures, fuel cycles, fuel designs, business strategies, and overall goals when approaching a project. Soon, the project notionally matures, having pinned down most choices based on the political, public, and economic issues that only a human can analyze. Next, focused design work begins, with reactor physicists performing neutronic calculations, thermal hydraulicists exploring coolability and transient behavior, mechanical engineers studying structural components, businessmen attracting investors and customers, and politicians pulling any necessary strings. Here, the designers are faced with

another nearly-infinite set of parametric design values. Evaluations at this level involve finding balances between physical constraints that allow favorable economics, excellent safety, and environmental responsibility — quantities that no longer require strictly human intelligence to determine. Computer algorithms, with their endless patience and attention, become invaluable tools in searching for, locating, and implementing the design changes that lead to the best performance. This research investigates a collection of methods working together to assist engineers in rapidly designing and optimizing advanced nuclear reactors.

## 1.1 Primary Considerations in Reactor Design

An ideal nuclear reactor must be safe, sustainable, economical, and proliferation resistant. When Fermi first demonstrated a chain nuclear reaction, he wondered with his colleagues whether or not humanity would ever allow such an amount of radiation to be produced to generate electricity [1]. History has shown that the public does in general accept nuclear energy, with 440 commercial power reactors in operation worldwide [2]. However, the major accidents in Chernobyl and Fukushima underline the importance of safety, and severely damage the otherwise excellent safety record of nuclear energy. Reactor designers have a responsibility to create systems that are extremely resilient to even beyond design basis events, since not all events are predictable. The concept of passive safety first demonstrated at Experimental Breeder Reactor Unit 2 (EBR-II) in April, 1986 [3] is a clear step towards this required resilience. Since then, passive safety features have been proposed or implemented in many reactor concepts, to some degree including the under-construction Westing-

house AP1000 plants in the Zhejiang and Shandong provinces of China, and at the Vogtle and Summer plants in the USA [4]. These do not offer indefinite passive decay heat removal, but they do offer a substantial buffer over reactors such as those at Fukushima.

Sustainable energy is that which is capable of powering humanity at our current and forecast levels for very long times without adversely affecting the environment. Though the precise amount of uranium available is uncertain, the most recent best estimates suggest that there are over 6.3 million tonnes of identified uranium ore worldwide, plus another 6 million tonnes of thorium in major deposits [5]. If the current demand of total energy (16 terawatts [6]) was conservatively held constant and nuclear energy was to provide half of it, these uranium resources would last for on the order of a mere 20 years under the current once-through fuel cycles. Clearly, nuclear reactors must attain higher total burnup to become truly sustainable, whether through a closed fuel cycle or a high-burnup once-through cycle. In advanced reactors under full recycling, these same resources could last for nearly 4000 years.

No reactor will be built en masse if it is not economically competitive with other energy sources. With new extraction technologies in use, the costs of natural gas and coal are projected to remain low for some time. These sources carry an uncertain hidden cost due to environmental damage. This, plus an eventual reduction of supply, may drive their prices up in the medium-term. Nuclear reactors are capital intensive, being more complicated and physically protected than conventional energy sources. However, once built, their fuel costs are very low thanks to the extraordinary energy density of nuclear fission. Additionally, nuclear waste is carefully packaged and

tracked, unlike waste from fossil fuels.

The inherent tie between nuclear reactors and nuclear weapons lies exclusively in the production of special nuclear materials. This tie manifests for two basic reasons: nuclear reactors can be used to breed chemically-separable fissile material (such as plutonium-239 or uranium-233) and the technology in the enrichment plants that prepare nuclear fuel can be exploited to enrich material (such as uranium-235) to weapons-grade. Both of these methods have been successfully used since the Manhattan Project to produce nuclear weapons. Proliferation arguments were a large reason the US abandoned its fast reactor program, and are a large part of the arguments against the allegedly-civilian nuclear program in Iran. Another concern is that highly-radioactive used nuclear fuel might be dispersed with conventional explosives in a dirty bomb. Therefore, an ideal reactor-fuel cycle system would minimize enrichment, production of fissile material, chemical separations, and high-level waste.

There are obviously competing factors that determine the ideal reactor. The challenge of the reactor designer is to balance cost with safety and sustainability with nonproliferation, as well as all the other combinations. To date, no reactor concept has been developed that is universally accepted as superior in all categories. But reactor designers have not stopped innovating. To assist in their quest for the ideal nuclear reactor (and to further optimize current designs), tools that enable rapid analysis of advanced concepts will be appreciated.

## 1.2 Motivations for High-Burnup Fast Reactors

Conceptual reactors that achieve very high burnup have unique capabilities and have been studied for some time. In particular, the breed-and-burn, or Traveling Wave Reactor (TWR) concept has roots in the 1950's, when Edward Teller took note of work by Feinberg [7]. This concept starts with an enriched core, irradiating a fertile blanket as it operates. By the time the initial fuel reaches its operational limit, the blanket has bred enough fuel to continue operation without separations or refabrication. Thus, without leaving the core, fuel that was once blanket becomes the primary fuel while fresh fertile material farther away becomes the low-flux tail of the eponymous wave. This enables up to a 40x improvement in resource utilization over traditional reactors, a reduction and potential elimination of enrichment requirements, and does not call for chemical separations as do other highly-sustainable concepts such as the Integral Fast Reactor (IFR). Furthermore, these simplifications reduce the fuel cycle costs.

The breed-and-burn concept has received substantial study since the late 1970's, when a concept that combined a moderated driver region with various configurations of blankets called the Fast Mixed-Spectrum Reactor (FMSR) was investigated [8, 9]. This reactor could theoretically propagate a wave (meaning, run indefinitely on fertile material without reprocessing) with a modest peak burnup of 20%. Teller regained interest in the concept and investigated autonomous, gas-cooled, non-shuffled, underground versions as reactors with manifest safety [10]. Other autonomous reactor designs followed, mostly focusing on monolithic designs similar to the CANDLE reactor [11].

The TWR has a threshold burnup that it must reach in order to attain its goal of propagating the breed-burn wave on fertile material, and this has been studied in detail [12]. For practical manifestations, this threshold is well beyond any burnup that has been achieved in any similar reactor. However, the benefits listed are substantial improvements in the majority of the categories of an ideal reactor system, and have inspired substantial study since their inception. More recently they inspired the creation of TerraPower, LLC, which is dedicating most of its resources to the development of a practical TWR [13]. Parallel efforts are underway, with core designers investigating ways to minimize the required burnup and a materials and fuels development team attempting to create and test fuel and structural systems capable of reaching the determined minimum.

High burnup is not easy to achieve. When fissions occur in solid nuclear fuel, the fission products are emitted with very high energy and cause substantial damage to the fuel material. Around 10% of fissions result in the production of noble gases xenon and krypton that internally pressurize the fuel. As a result, the fuel swells and otherwise deforms as it encounters neutrons. Solid fission products increase the fuel volume by around 1.18% per percent burnup [14]. Lanthanides chemically interact with the cladding, causing damage through fuel-cladding chemical interaction (FCCI). These fuel performance constraints from strain and corrosion limit the burnup in most TWR designs. The buildup of fission product poisons also neutronically limits the burnup for a given fuel enrichment. However this is much more constraining for thermal reactors, where the capture cross sections of some nuclides are very high, than it is for fast reactors. High burnup is typically accom-

panied by high doses to the structural materials (cladding, duct, etc.). Non-fissile materials undergo property changes with irradiation as well. When structure grows and distorts, it can cause cooling and/or mechanical problems. Light-water reactors typically burn their fuel to an average of 60 MWd/kg ( $\sim 6\%$  fissions per initial heavy metal atom (FIMA)). The burnup and dose records in fast reactors are held by EBR-II and Fast Flux Test Facility (FFTF) which irradiated fuel to 200 MWd/kg and structural steel to just above 200 displacements per atom (DPA) [15, 16].

Designing high-burnup fast reactors such as the TWR requires unconventional analysis and considerations. Primarily, the burnup that must be achieved to successfully propagate a uranium-plutonium interface wave in a TWR is a strong function of fuel assembly design. To reduce void worth, sodium-cooled fast reactor (SFR) designers typically shorten the core to enhance axial leakage. In a TWR, the extra leakage may require more burnup and more DPA, and thus be detrimental to the materials and fuel. Regardless of material issues, if the required wave-propagation burnup is above the burnup threshold of fission product poisoning, the reactor is infeasible. Care must constantly be taken to understand the effects of any design change on the required burnup. Even small changes to clad or duct thickness amplify through the equilibrium cycle of a TWR to sensitively affect this parameter. Without this added consideration, nuclear reactor design optimization is a daunting task, but with it, we are strongly motivated to create applicable systematic optimization methods. Furthermore, the behavior of high-burnup fuel requires explicit modeling of fuel performance, including the reactivity effects of fuel growth, fission gas release, and the squeeze-out of sodium bond.

The sensitivity of TWRs necessitates high-fidelity modeling, which in turn inspires a desire to perform high-fidelity calculations quickly. One well-known approach to speeding up calculations is the application of perturbation theory.

### **1.3 The Evolution of Perturbation Methods for Reactor Design**

The concept of approximating solutions of complicated equations based on perturbations of simpler or previously solved systems has been widely applied to physical problems since Newton's work on the orbits of the Earth-Sun-Moon system. Since then, perturbation theory has been applied to a vast quantity of problems, most notably to quantum mechanics, where the few analytic solutions of the Schrödinger equation can be expanded to demonstrate more complicated phenomena such as the Stark and Zeeman effects. This experience enabled the early nuclear physicists to apply perturbation theory to the analysis of nuclear reactors, who used it to understand fundamental behavior of chain reactions, such as the effect of a burst of neutrons on a critical reactor and the change in reactivity given small changes in nuclear properties [17].

#### **1.3.1 Variational Methods**

Closely related to perturbation theory are variational methods, generally attributed to Roussopolos [18], which provide the means to solve perturbed systems while introducing only second-order errors. Functionals of the flux, such as reaction rates and resonance integrals, are among the quantities which can be well-

approximated without full recomputation. Variational methods have been heavily developed and generalized. Pomraning, Usachev, and Gandini built on Roussopolos' work to develop a variational principle to approximate functionals of eigenfunctions in homogeneous and fixed-source problems [19, 20, 21, 22]. These methods of Generalized Perturbation Theory (GPT) formulate a general functional by appending the unperturbed system equations to an objective function with Lagrange multipliers. Reordering terms by introducing the adjoint operators of the homogeneous equations forces first-order variations to vanish. It can then be systematically shown that the functional is identical to the objective with no perturbations and that first-order perturbations lead to second-order variations in the approximation. Stacey extended this progress to include expressions for ratios of linear functionals of the flux and bilinear functionals of the flux and adjoint flux, applying it to estimate reactivity worth and reaction rate ratios [23].

### 1.3.2 Explicit High-Order Perturbation Theory

Reactor design optimization fits well with explicit perturbation methods that compute the perturbed real and adjoint fluxes instead of perturbed functionals. Gandini developed an explicit high-order perturbation theory [24] that was first applied by Palmiotti [25]. This method, termed the "Standard Method" due to its relation to classic quantum mechanics, involves expanding the perturbed flux and perturbed eigenvalue as an infinite sum of terms, with each expansion term the solution to a recurrent system based on a similar problem in quantum mechanics due to Mitani [26]. The recurrent system is solved by expanding each term in the real flux harmon-

ics, multiplying by the adjoint flux harmonics, and making use of the orthogonality of these bases to reduce the expansion to a single term. The flux harmonics are computed using the low-order mode sweeping method, which sequentially subtracts contributions of lower-order harmonics at each outer iteration of the power method.

The original explicit formulation was able to match very large perturbations in one-group problems, but did not perform well for multigroup problems, apparently due to the incompleteness of the eigenfunction basis as computed by the power method [27]. It was modified by replacing the full-spectrum harmonic expansions by individual expansions for each energy group [28]. A pseudo-orthogonality relationship for the group fluxes was found to be satisfied for cases studied, but was not proven mathematically. These modifications led to satisfactory performance in 2-D cases with up to three energy groups, considering up to 25 flux harmonics.

Another related method is the modal-local method [29, 30], where the shape function of the flux was treated as the sum of a modal shape function and a local shape function. The modal function is constructed on a basis of  $\lambda$ -eigenmodes with the first several harmonics, while the local function is evaluated for any given local perturbation by a fixed source problem. Only on the order of ten harmonics are required to model large perturbations because the higher-order localized aspects are treated by the local function. The method was used to solve the space-time inverse kinetics problem to determine the control rod worth of a gas-cooled test reactor.

### 1.3.3 Time-Dependent Perturbation Theory

Perturbation methods that determine time-dependent effects of design or data perturbations have been developed under the name Depletion Perturbation Theory (DPT). Gandini pioneered 0-D DPT by implementing GPT variational methods on the nuclide burnup equation. He applied the methods of standard GPT to the depletion problem, deriving new adjoint functions and discovering that solutions to the adjoint nuclide depletion equation can be used as nuclide importance functions. Doing so, he showed several examples of calculating the sensitivity of nuclide number densities to changes in the terms of the transmutation matrix [31]. However, the analysis of the depletion was completely decoupled from the neutron transport equation, though the neutron and nuclide fields are tightly coupled in nature.

Following Gandini, Kallfelz, et al. applied static and time-dependent GPT to burnup problems, focusing on the effects perturbations in the flux distribution have on the plutonium isotopic evolution [32]. They considered the coupled influences of the nuclide-neutron systems by computing the change in flux for each change in number density with static GPT. In a 1-D spherical reactor model with parameters typical to a liquid-metal cooled fast breeder reactor (LMFBR), perturbation calculations were compared against direct calculations, confirming the validity of the derivation to reasonable accuracy.

Working with the same team as Kallfelz, Gandini noted that DPT methods can be used not only to estimate sensitivities due to cross sections, thereby making a priority list of which measurements are most important, but also to determine how the concentration of various actinides affects the actinide inventory at end of cycle

(EOC). They suggested to use DPT as a method useful in the strategic reduction of transuranic material [33].

Williams and Weisbin did parallel work at ORNL, looking specifically at sensitivity and uncertainty of cross sections [34]. They derived the adjoint nuclide equations from first principles, using a variational method rather than the analogies to the neutron transport equation previously used by Gandini, introducing a new rigor into DPT, and created ORIGEN-A to solve them based on the well-known burnup code. With the variational principle in hand, they showed the congruence with the neutron adjoint, and showed that the physical interpretation of the adjoint nuclide field can be similarly treated as an importance weighting (the same conclusion Gandini had made), following a principle of conservation of nuclide importance. They suggest applications to nuclide channel theory, whereby nuclide paths can be studied as atoms are transformed from their initial compositions to their final compositions.

Noting a lack of a unifying manifestation of previous work on DPT, Williams rigorously developed quasi-static coupled depletion perturbation theory [35]. Dissatisfied with Kallfelz's treatment of the nonlinear interactions between neutron and nuclide fields, he derived coupled treatments based on the assumption that the quasi-static approximation (where neutron flux is assumed constant over a finite burn-step during which nuclide densities change) is accurate. White took these new theoretical achievements in DPT and built a code based on the VENTURE/BURNER system to demonstrate their applicability to more realistic problems [36]. Downar and Yang applied Williams' theory to equilibrium cycles operating under constant power, using the average flux between time nodes rather than the quasi-static assumption

[37]. This work was later extended to consider sensitivities of closed equilibrium cycles [38]. Van Geemert developed a variational treatment of equilibrium fuel management and applied DPT to minimize the feed enrichment in a 1.5 group, 2-D light-water reactor (LWR) equilibrium problem [39].

## 1.4 Nuclear Reactor Design Optimization

Nuclear reactor design optimization has been ongoing since the first reactor was conceived, but computational tools to accelerate the obvious practice of running direct parameter sweeps surfaced later. For example, Gandini, et al. [40] used GPT to drive optimization, showing an application that produced a beryllium distribution to optimize the Doppler effect in a small fast reactor. They used linear programming techniques to perform the optimization. Since then, many other similar applications of GPT have been pursued. GPT and DPT have been shown to effectively capture the effects of small perturbations in cross sections and other input on arbitrary functionals in time-dependent simulations. In these methods, adjoint calculations are executed for each particular functional (representing for example, the reaction rate in a certain region) whose response to many perturbations may be evaluated efficiently. These methods are powerful for understanding the effects of data and model uncertainties, or for performing optimization of a very specific quantity.

Choosing a different approach, Tzanos, et al. applied a Taylor series expansion of the diffusion equation around an operational state of a one-dimensional cylinder reactor simulation, implementing linear programming on it to optimize reactor parameters subject to a power peaking constraint [41]. An iteration was developed

where each step in the optimization was followed by another series expansion.

A third optimization approach appeared when Terney and Williamson [42] used the Pontryagin Maximum Principle to optimize partial reload problems in realistic LWR simulations. They appended state equations to an objective function with Lagrange multipliers and determined corresponding Euler-Lagrange equations by forcing first-order variations of the resulting functional to vanish. Repeatedly solving the system and Euler-Lagrange equations, they determined optimal reload patterns. The derivation of this approach has close ties to that of the variational perturbation theory that underlies GPT, but in this case, the stationary conditions provided by the solution of the adjointed functional are designed to seek the optimal design. The optimal control approach later underwent substantial development. Drumm and Lee [43] extended the method to use a penalty function treatment of the power peaking inequality to optimize the cycle length of a LWR based on the burnable poison distribution. Wu [44] approached the problem by including the power peaking inequality constraint in the functional using a direct adjoining concept from optimal control theory [45]. Sorensen [46] applied this direct adjoining control theory to a 2-D, two-group LWR optimization problem, implementing backwards diffusion theory to account for inequality violations that can occur in previous implementations. Building on this, Davis [47] applied the theory to a 3-D, multigroup fast reactor problem with microscopic depletion.

A very large body of work has been produced in the stochastic optimization of reactor design and reload design, typically implementing genetic algorithms or simulated annealing [48, 49, 50]. Such methods use natural analogies to intelligently

propagate an otherwise random set of design parameters until improvement ceases. The high popularity of these methods stems from their simplicity of implementation — any existing simulation can become the driver of a stochastic optimization system without modification. As such, the methods are constrained in speed to the speed of the existing drivers, which are typically direct diffusion-depletion solvers. Without the acceleration offered by perturbation-based methods or the derivative information available to control-theory methods, stochastic optimization can be very slow. This issue has been largely tackled through computational parallelism.

## 1.5 The Objective and Organization of this Thesis

Analyzing nuclear reactor performance after many cycles, such that it exhibits its long-term behavior is essential in this work. Safety performance of uranium-fueled high-burnup fast reactors are almost always most constraining at these equilibrium conditions (or at end of life (EOL)), primarily because plutonium nuclides have replaced uranium nuclides as the primary fuel. The delayed neutron fraction  $\beta$  from Pu-239 fission is less than half that from U-235 fissions, enabling smaller perturbations to bring the reactor closer to prompt-supercritical. Often, reactors that appear to be inherently safe at beginning of life (BOL) are not so at equilibrium. Accordingly, coolant voiding and transient overpower initiators are more difficult to handle. Furthermore, fuels and materials limits on burnup and fluence are challenged at their extremes, which in most cases occur at equilibrium as well. The objective of this research is to develop a method that allows the efficient system optimization of a reactor at equilibrium considering many design parameters and metrics.

Rather than relying upon DPT, optimal control, or direct stochastic optimization, a new approach is explored in this research. DPT requires forward calculations for each reference state and adjoint calculations for each integral quantity to be optimized. Its advantage is that once these are computed, the effect of arbitrary design perturbations on the single integral quantity may be efficiently determined. However, considering system optimization requires the sensitivities of many such responses, and therefore requires many such adjoint calculations. To bypass this inefficiency, this research develops an explicit perturbation method that determines the perturbed flux rather than only its effect on a particular integral.

Clearly, an efficient manner to compute the equilibrium state of a nuclear reactor is essential. To this day, many equilibrium analysis routines, including the EQL3D [51] routines developed within the ERANOS framework, reach the equilibrium state by running explicit calculations of one cycle after the other. While the convergence varies with reactor design, it regularly requires hundreds of full-core calculations. The one outstanding exception in the literature is the REBUS-3 code [52], which contains a fast equilibrium method that offers a substantial gain in computational efficiency. The method therein is reviewed in Chapter II, followed by a demonstration of some deficiencies of the method when applied to high-burnup fast reactors. An Enhanced Equilibrium Cycle (EEC) method is then proposed and demonstrated to improve upon these deficiencies as an extension to the existing method. EEC is implemented within the Advanced Reactor Modeling Interface (ARMI), a recently-developed code system that provides multiphysics coupling and automation [53]. The built-in abilities of ARMI to automatically calculate thermal-hydraulics, fuel

performance, transients, economics, and mechanical performance are relied upon in later chapters to evaluate complex objective functions during optimization, and are also introduced in Chapter II.

The EEC method, while much faster than direct cycle calculations, still takes considerable time for high-fidelity models. For design optimization, speed is of the essence. To cover a large design space, very many EEC calculations must be evaluated. Faster methods enable more design points to be evaluated, and therefore allow a more complete optimization. An explicit Modal Expansion Perturbation Theory (MEPT) method is developed in Chapter III that determines changes in real and adjoint flux distributions under large perturbations, on the order required for optimization studies. The method utilizes large numbers of  $\lambda$ -mode flux harmonics of the steady-state neutron balance equation as calculated by a Krylov-based iterative eigenvalue solver. Krylov-based iterative solvers have been shown to converge more rapidly than the power iteration in some situations [54], but these advantages are not focused upon, since only a single reference calculation of the flux harmonics is required for a more numerous set of perturbed flux calculations. MEPT is coupled to EEC in Chapter IV, forming an Equilibrium Cycle Perturbation Theory (ECPT) that propagates design perturbations through time with coupled neutronics and depletion, determining the equilibrium manifestation of BOL design perturbations. As mentioned previously, since the system is developed within the ARMI, many important metrics of reactor design are automatically calculated for each perturbed equilibrium state.

In Chapter V, ECPT is implemented as the kernel of a multidisciplinary design

optimization (MDO) problem. It is run many times around a particular reference state, building a large set of independent variables and their resulting dependent variables, including maximum burnup, equilibrium  $k_{eff}$ , the coolant density coefficient of reactivity, total fuel cost, and the delayed neutron fraction. These data are used to perform a multidimensional regression using the Alternating Conditional Expectation (ACE) algorithm [55], resulting in a case-dependent surrogate model of the trained data. Using the surrogate, user-friendly multiobjective optimization is accomplished through the Physical Programming paradigm [56]. The surrogate is used during the optimization, and can be reused with different objective functions as determined by the user. In this way, a designer may explore their optimization objectives without re-running the multitude of ECPT cases on the design space multiple times. An example optimization of a high-burnup reactor is performed.

## CHAPTER II

### An Enhanced Equilibrium Cycle Method

The equilibrium cycle of a nuclear reactor is the state that is attained after the system has been operated in a repetitive manner for a very long time. For example, in a traditional fast reactor, natural uranium may be fed into a reprocessing plant and combined with the reprocessed discharge of a reactor to fabricate the new charge fuel for each next cycle. The discharged material from the first cycle (when the fresh core is loaded only with enriched uranium) will be different from the discharge of the second cycle — more plutonium and minor actinides will begin to build up. However, as multiple cycles are operated, the charge and discharge vectors will converge to particular, repeated values. This is the equilibrium condition. It determines the long-term performance of the reactor, and is therefore an essential point of analysis for any reactor design.

Design constraints are typically active at equilibrium. The fuel and materials will reach their highest burnup and dose, and therefore their limiting levels of corrosion, internal pressure, strain, and distortion. The safety characteristics in many fast

reactors tend to be most limiting at the equilibrium condition due to the low delayed neutron yield of plutonium and the fact that plutonium concentration is usually highest at equilibrium.

Considering the importance of the equilibrium cycle, there are surprisingly few computer codes that compute the equilibrium state for fast reactors. Argonne National Laboratory's REactor BUrnup System (REBUS) [52] is the most prominent, followed by the EQL3D routines in the CEA's ERANOS [51]. Of these, the REBUS methodology is most interesting because of its specialized equilibrium iteration that can arrive at results very rapidly, whereas EQL3D computes explicit cycles until convergence.

The fast equilibrium methodology from REBUS is an invaluable tool for rapidly designing equilibrium cycles. However, its method is shown in this chapter to be inaccurate in high-burnup reactors such as the TWR. This chapter demonstrates this breakdown of accuracy and develops an EEC method that produces improved results.

## **2.1 The Traditional Equilibrium Cycle Methodology**

The REBUS equilibrium cycle search methodology is well-documented [57], and is summarized here for completeness. For comparison, the standard direct method of reaching equilibrium conditions involves first computing the flux and depletion of the first cycle of a fresh core, followed by a fuel management operation. The process is repeated, with one flux calculation and one depletion calculation per cycle. At each step, each assembly accumulates a single cycle's worth of burnup. Eventually,

all the fuel assemblies that were in the fresh core will be discharged and the core will settle into its equilibrium state and converge. In strong contrast to this, the REBUS equilibrium search starts with a single global flux guess and, using this single guess, depletes each charge assembly  $T$  times in a row, where  $T$  is the number of cycles that the assembly will reside in the core. The full depletion history of each assembly is computed at each step. To accomplish this, the method computes the beginning of equilibrium cycle (BOEC) number densities of each fuel region as the end of equilibrium cycle (EOEC) densities of the region that shuffles into it at the next cycle. Starting with the charge fuel, compositions of all fuel regions in a core are computed sequentially along their fuel management paths. Once these densities are computed throughout a core, the flux is updated and the process continues. In this method, the entire burnup distribution of the core is estimated after a single flux calculation, whereas the direct method would take at least  $T$  flux calculations to approach a reasonable estimate. The equilibrium iteration converges to the equilibrium condition much faster than the direct method.

The REBUS equilibrium methodology includes much flexibility in the number of fuel regions to consider, with assemblies possibly being averaged into larger groups for coarse spatial meshes that were required for reasonable run times on early computers. For simplicity, we review the equilibrium derivation under the assumption that each spatial region is comprised of a single axial block of a single assembly, and that each assembly is shuffled at each cycle.

### 2.1.1 Mathematical Representation of the Reactor

A reactor is made up of  $K$  spatial regions ( $r_1, r_2, \dots, r_K$ ) each characterized by an atom density vector. When an assembly enters the core, it is considered to be at stage one, or  $\tau = 1$ . After one cycle, it is at stage two ( $\tau = 2$ ). When the material in region  $r$  is scheduled for discharge, it is at stage  $\tau_d$ . The index tuple  $(\tau, r)$  fully defines a particular composition during a burn cycle. Under our simplifying assumption that each assembly moves at each cycle,  $\tau_d = K + 1$ , and only the  $r$  index is required to identify a material.

The number density of a composition containing  $I$  active nuclides after  $r$  burn cycles is given by the region density vector  $N_r(t)$ . As the reactor operates, the initial compositions will undergo nuclear reactions to transmute into new compositions depending on the neutron flux. The nominal time-dependent behavior of the reactor is determined by these transmutations, which are described by the Bateman equation stating that the time rate-of-change of each nuclide  $i$  is

$$\frac{\partial N_i}{\partial t} = -N_i \left[ \sum_{g=1}^G \sigma_{a,g}^i \phi_g + \lambda_i \right] + \sum_{j=1}^I N_j \left[ \left( \sum_{g=1}^G \gamma_g^{j \rightarrow i} \phi_g \right) + \lambda_{j \rightarrow i} \right], \quad (2.1)$$

where

$\sigma_{a,g}^i$  is the  $g$ -th group absorption cross-section for nuclide  $i$ ,

$\lambda_i$  is the decay constant for nuclide  $i$ ,

$\gamma_g^{j \rightarrow i}$  is the transmutation yield from nuclide  $j$  to nuclide  $i$ , and

$\lambda_{j \rightarrow i}$  is the decay constant from nuclide  $j$  to nuclide  $i$ .

This system of equations can be formulated into matrix notation as:

$$\frac{\partial N}{\partial t} = AN, \quad (2.2)$$

where the diagonal terms of the transmutation matrix  $A$  contain the first term, representing losses in Eq. (2.1) and the off-diagonal terms contain the second term, representing gains. The general solution of this transmutation equation is a matrix exponential

$$N(t) = e^{\int A dt} N_0, \quad (2.3)$$

where  $e^{\int A dt}$  is often condensed to  $B(t)$  and referred to as the burn matrix. The matrix exponential is formally defined as the infinite sum

$$e^X = \sum_{k=0}^{\infty} \frac{1}{k!} X^k, \quad (2.4)$$

but is typically evaluated by truncating the sum after a certain number of terms have been computed, although alternate solution methods do exist [58].

Let the state of the reactor at the beginning of a burn step be described by the set of density vectors  $\{N_r(t_0)\}$ . Then the time dependence of the reactor is given by the following equations:

$$\frac{d}{dt} N_r(t) = A_r[\phi(t)] N_r(t)$$

$$F [\phi (t) ; N_1 (t) , N_2 (t) \dots , N_K (t)] = 0 \quad (2.5)$$

The diffusion operator  $F$  explicitly shows its parametric dependence on all region densities. Since the flux is dependent on the number densities and the number densities are dependent on the flux, the behavior of the reactor is the solution of a coupled system of equations. Fortunately, the rates of change of the number densities is generally very slow, and the quasi-static approximation is accurate. Under this assumption, the solution to the depletion equations is the matrix exponential

$$N_r (t) = e^{A_r[\phi(t_0)(t-t_0)]} N_r (t_0) = B_r N_r (t_0) . \quad (2.6)$$

This model of a coupled neutron-nuclide multi-region nuclear reactor provides the starting point for the equilibrium cycle methodology.

### 2.1.2 Formulation of the Equilibrium Cycle

The equilibrium cycle calculation involves the repetitive application of the following steps:

1. Burn Cycle.
2. Discharge
3. Reprocessing
4. Fuel Shuffling
5. Fuel Loading

The initial state of the reactor is described by the number density set  $\{N_r^{(0)}\}$ , where the superscript represents the iteration number. The first burn step of an equilibrium calculation involves computing the global flux and generating burn matrices  $B_r^{(1)}$ , and then depleting each stage of each region to its next stage using these new region-dependent burn matrices ,

$$N_{r+1}^{(1)} = B_r^{(1)} N_r^{(0)}. \quad (2.7)$$

The discharged nuclides are subject to the external cycle model:

$$\bar{N}^{(1)} = Q_r (N^{(1)}) N^{(1)} + Q_f (N^{(1)}) N_f, \quad (2.8)$$

where  $Q_r$  represents losses during reprocessing and  $Q_f$  represents gains from an external feed, and  $\bar{N}^{(1)}$  represents the feed material for the next cycle, which is also the discharge from the external cycle. In general, the shuffling step may or may not require spatial repositioning, but always requires an increment in stage number, representing movement through time. In this simplified derivation, shuffling always results in movement to the next region. The final step is to load the new charge material densities,  $\bar{N}_K$ , into the first region. As a result, the state of the reactor before the second equilibrium iteration is the set of atom density vectors,  $\{\bar{N}_1^{(1)}, B_1^{(1)} \bar{N}_1^{(0)}, B_2^{(1)} \bar{N}_2^{(0)}, \dots, B_K^{(1)} \bar{N}_K^{(0)}\}$ . The first vector is the freshly charged fuel, the second vector is the once-burned material, and so on. Each subsequent  $B$  matrix can be written in terms of a cascading set of the previous  $B$  matrices. In other words, the discharge number density vector can be related to the charge vector ( $N_1$ ) by a

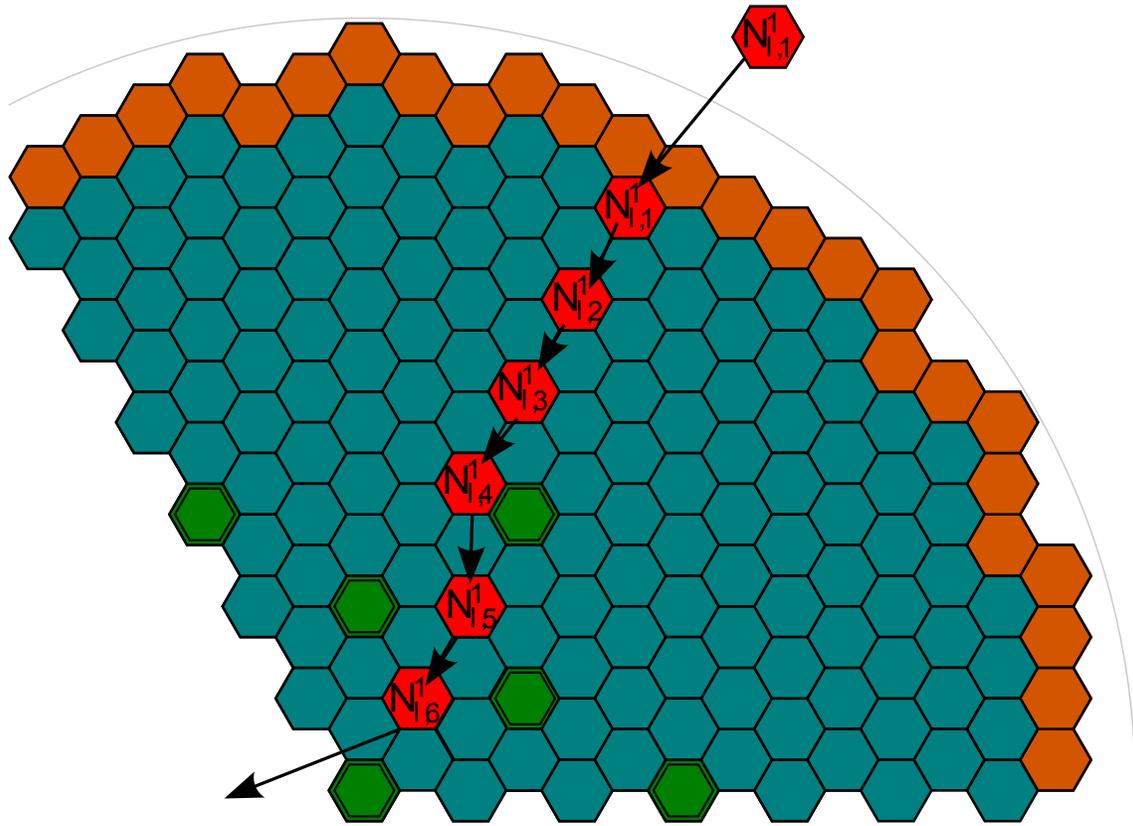


Figure 2.1: A simple visualization of a single chain in a repetitive fuel management scheme. At each cycle, one assembly is charged, one is discharged, and the rest cascade towards the center. All number densities in this chain can be determined by applying a sequence of burn matrices to the charge densities with a single global flux iterate.

series of multiplications of the proper burn matrices:

$$N_K = B_K B_{K-1} \dots B_1 \bar{N}_1. \quad (2.9)$$

A simple visualization of a single chain of material converging through a reactor model is shown in Figure 2.1. In an  $N$ -batch core,  $\frac{1}{N}$  of the fuel is removed at each cycle.

One can establish a functional form for the new charge vector based on the in-core and external cycles where the new charge vector is a function of the cycle length  $T$ , the charge enrichment  $e$ , and all preceding charge vectors,

$$\bar{N}^{(\mu+1)} = f(T^{(\mu+1)}, e^{(\mu+1)}, \bar{N}^{(\mu)}, \bar{N}^{(\mu-1)}, \dots, \bar{N}^{(\mu-K+1)}). \quad (2.10)$$

The burnup, fluence, DPA, etc. at this  $(\mu + 1)$ th burn cycle can be similarly indicated by a functional form of the cycle length, enrichments, and all preceding charge densities as:

$$b^{(\mu+1)} = b(T^{(\mu+1)}, e^{(\mu)}, \bar{N}^{(\mu)}, \bar{N}^{(\mu-1)}, \dots, \bar{N}^{(\mu-K+1)}). \quad (2.11)$$

Similarly, the unpoisoned multiplication factor can be computed at some fraction  $\alpha$  through the burn cycle as

$$k^{(\mu+1)} = k(\alpha T^{(\mu+1)}, e^{(\mu)}, \bar{N}^{(\mu)}, \bar{N}^{(\mu-1)}, \dots, \bar{N}^{(\mu-K+1)}). \quad (2.12)$$

The enrichment is not time-independent with respect to the burnup or  $k_{eff}$  because it is a function of  $\bar{N}^{(\mu)}$ , the preceding charge density vector. If the new charge vector and all preceding charge vectors were equal, or

$$\bar{N} = f(T, e, \bar{N}, \bar{N}, \dots, \bar{N}) \quad (2.13)$$

where  $T$  and  $e$  are the cycle length and charge enrichment, and if the burnup  $b_0$  was as desired, then the reactor would be operating in an equilibrium condition. In this

condition, several quantities lose their dependence on the cycle index  $\mu$ :

- The burn time and enrichments
- The charge and discharge vectors
- The transmutation operators in the in-reactor cycle

Given this condition and the converged burn matrix for each stage, one may obtain all number densities at all times simply by applying the proper sequence of transmutation matrices. Compactly, one may form a matrix equation where:

$$B(T, e, \bar{N}) = \Pi_{r=1}^K B_r(T, e, \bar{N}), \quad (2.14)$$

which can be used to write a full-core equilibrium mapping of charge densities  $\bar{N}$  to discharge densities  $N$ ,

$$N = B(T, e, \bar{N}) \bar{N}. \quad (2.15)$$

If a reactor is in a state where the charge and discharge from the core are identical for all burn steps, the reactor is said to be in a cyclic mode. If the external cycle is also considered and converged, then the reactor system is said to be in equilibrium.

Overall, REBUS-3 defines 4 basic states of equilibrium:

1.  $N = B(T, e, \bar{N}) \bar{N}$  cyclic mode
2.  $b_0 = b(T, e, \bar{N})$  cyclic mode with a burnup constraint. Burn time  $T$  is adjusted to achieve burnup compliant with the constraint

3.  $\bar{N} = Q_r(N) N + Q_f(N) N_f$  unconstrained equilibrium mode. This gives the relation to the discharged nuclides, the feed nuclides, and the new charge (in effect, it is the external cycle, with reprocessing and refabrication)
4.  $k_0 = k(\alpha T, e, \bar{N})$  constrained equilibrium mode (satisfies specific criticality at time  $\alpha T$ )

### 2.1.3 Solution of the Equilibrium Cycle

With the equations that define the equilibrium cycle at hand, we now consider the computational iterations used to determine their solution. The flow of logic in REBUS-3 is shown graphically in the Figure 2.2. An initial enrichment is guessed and the unconstrained equilibrium is found, resulting in a burn time and a charge density vector. The charge enrichment is modified as necessary to bring  $k_{eff}$  to the desired multiplication constant.

#### 2.1.3.1 The Cyclic Mode

The cyclic mode returns an equilibrium transmutation matrix given a charge vector  $\bar{N}$ . One cycle is broken up into several subintervals to account for changes in flux magnitude through the cycle. Once a solution is converged upon, the burn matrices for each subinterval are recombined via matrix multiplication to get the effective single-cycle burn matrix for the next step in the cyclic mode iteration. When the burn matrix changes substantially with time or the cycle length is long, this method accommodates the coupled nature of the problem by breaking the cycle into more time nodes. The cyclic mode iteration is shown in Figure 2.3.

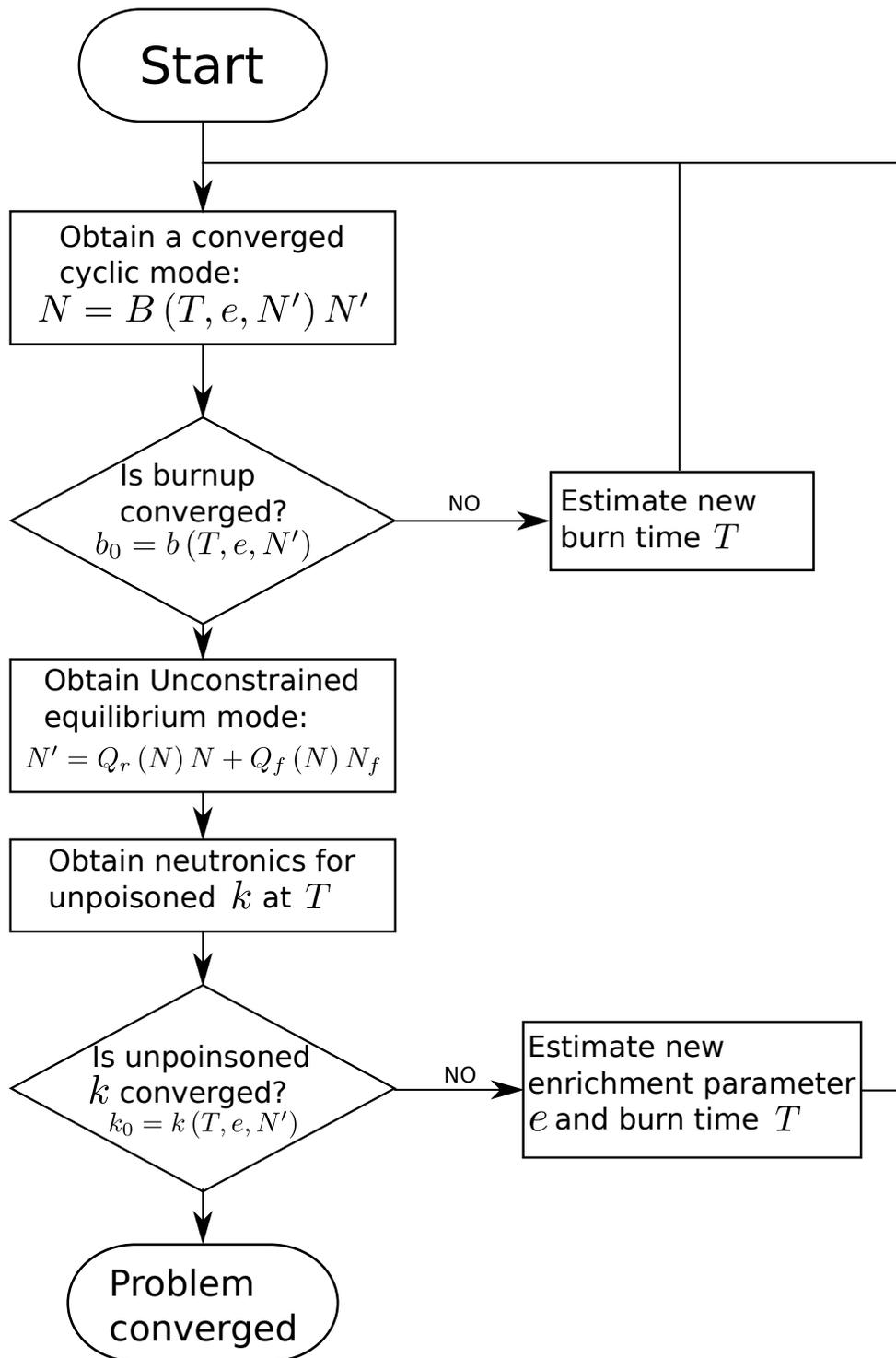


Figure 2.2: Flowchart of the REBUS-3 equilibrium cycle.

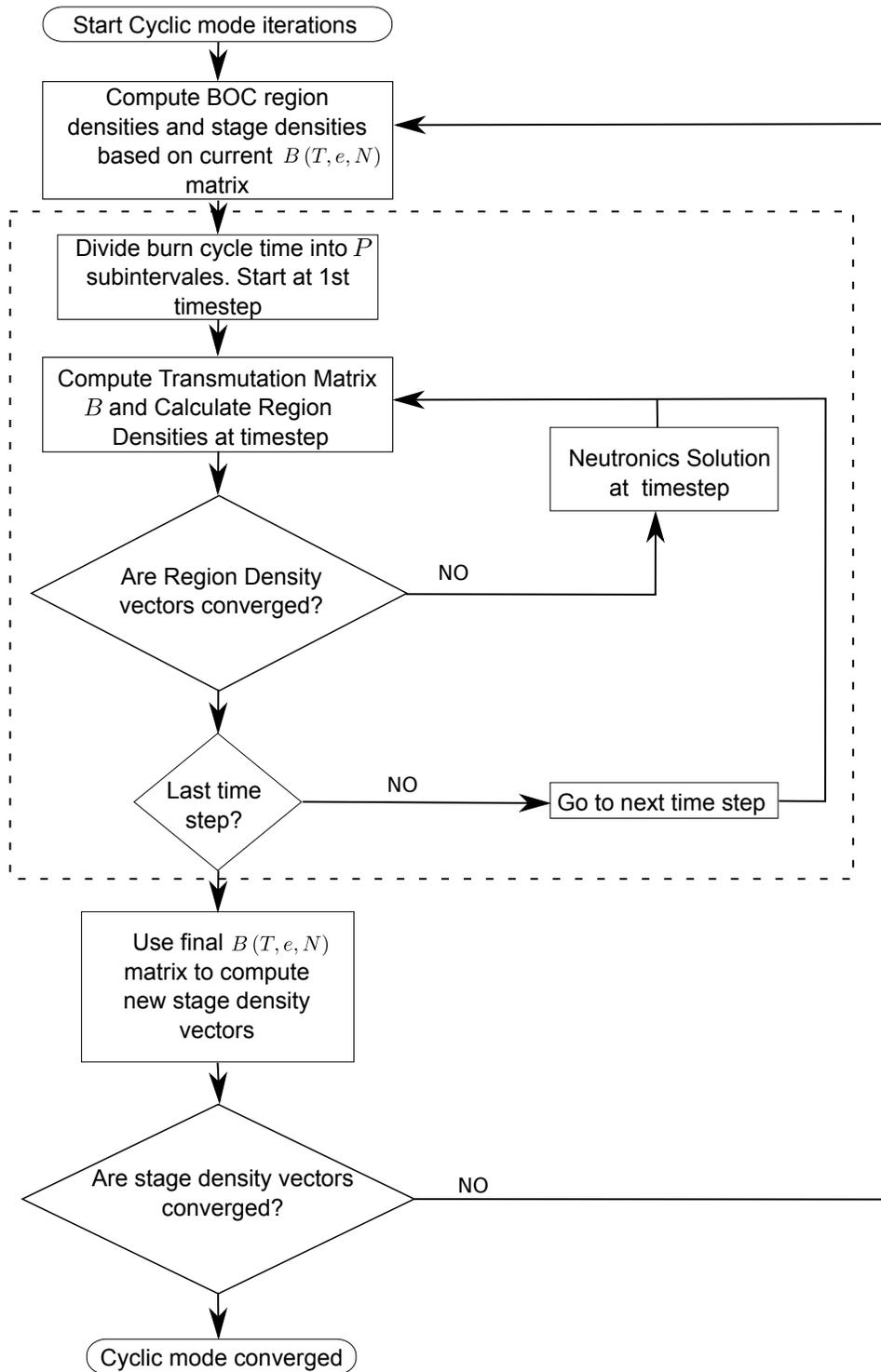


Figure 2.3: Flowchart of the REBUS cyclic mode iteration.

Note that any change in the burn time  $T$  or the charge enrichment  $e$  will alter the cyclic mode. The processes within the dotted lines are known as the region-density iterations and are responsible for computing the cycle-averaged burn matrix between each cyclic iteration. The calculation proceeds by dividing burn cycle time  $T$  up into  $P$  sub-intervals of equal length,  $t_p = ph$  where  $h = T/P$ . Then, the code must compute a converged  $B$  matrix over each of the  $P$  different subintervals. The coupled equations over a single subinterval are

$$\frac{d}{dt}N_r(t) = A_r^p[\phi(t)]N_r(t) \quad (2.16)$$

$$F[\phi(t_p); N_1(t_p), N_2(t_p) \dots, N_K(t_p)] = 0 \quad (2.17)$$

where  $p$  ranges from 0 to  $P$ . This system is solved iteratively for each sub-interval using an interval-averaged transmutation matrix. To obtain a converged region density vector at the next time node  $t_{p+1}$ , the following iterative sequence is performed.

### 2.1.3.2 The Region-Density Iterations

Given the density vector for each  $r$  at time  $t_p$ , set iteration counter  $q = 0$ , and iterate:

1. Compute the average transmutation matrix over the subinterval,  $\bar{A}^{p+1(q)}(t_p, t_{p+1})$
2. Using  $\bar{A}^{p+1(q)}$ , compute the burn matrix  $B^{p+1(q)}$  using a scaling and squaring technique (keeping all eigenvalues in the expansion  $< 1$  for rapid convergence)

3. Compute the region density vector  $N_r^{(q)}(t_{p+1}) = B^{p+1(q)} N_r(t_p)$
4. Using new region densities, compute the flux at  $t_{p+1}$
5. Using  $\phi^{(q+1)}(t_{p+1})$  compute the transmutation matrix  $A(\phi^{q+1}(t_{p+1}))$
6. Compute new averaged transmutation matrix for the subinterval as  $\bar{A}_{(t_p, t_{p+1})}^{p+1(q+1)} = \frac{1}{2} \{A[\phi(t_p)] + A[\phi^{(q+1)}(t_{p+1})]\}$
7.  $q = q + 1$  and continue

This iteration produces a sequence of density vectors for each region at time  $t_{p+1}$  until  $q$  exceeds a user-specified limit or until the region density vectors are converged to a user-specified criterion. This iteration provides so-called constant power depletion, distinct from and more accurate than the standard quasi-static depletion [37].

To start the region-density iteration, we guess that the subinterval-averaged transmutation matrix is equal to the beginning of cycle (BOC) transmutation matrix for the first subinterval ( $p = 0$ ):

$$\bar{A}_{(t_0, t_1)}^{p+1(0)} = A[\phi(t_0)] \quad (2.18)$$

For subsequent subintervals, we linearly extrapolate the transmutation matrices considering those at  $t_p$  and  $t_{p-1}$ .

$$\bar{A}_{(t_p, t_{p+1})}^{p+1(0)} = (3A[\phi(t_p)] - A[\phi(t_{p-1})]) / 2 \quad (2.19)$$

### 2.1.3.3 Obtaining an Unconstrained Equilibrium Mode

For a given enrichment vector  $e$ , the unconstrained equilibrium mode is found by determining the cycle length  $T = T(e)$  and the charge density  $\bar{N} = \bar{N}(e)$  such that we achieve a cyclic mode subject to a burnup constraint and the reprocessing and fabrication equation. The direct successive approximation is accelerated in REBUS-3 by ignoring the dependence of  $B$  on the cycle length and enrichment:

$$\begin{aligned}
 N^i &= B(T, e, \bar{N}) \bar{N}^i \\
 \bar{N}^{i+1} &= Q_r(N^i) N^i + Q_f(N^i) N_f \\
 N^{i+1} &= B(T, e, \bar{N}) \bar{N}^{i+1} \\
 \bar{N}^{i+2} &= Q_r(N^{i+1}) N^{i+1} + Q_f(N^{i+1}) N_f.
 \end{aligned} \tag{2.20}$$

Recall that  $Q_f$  depends on  $N_i$  primarily because it is built through a prioritization mechanism that is dependent on the amount of feed required to build the new charge. In this way, the effect of the external cycle on the charge vector is modeled.

### 2.1.3.4 The Constrained Equilibrium Mode

A cyclic mode subject to a criticality constraint is determined by varying either the charge enrichment or the cycle length and parabolically interpolating until the  $k_{eff}$  at the desired point in the cycle is close to a target value. Two levels of iteration are involved in a criticality search, each with a separate convergence criterion. Given a guessed cycle length or charge enrichment, an inner iteration determines a con-

verged cyclic mode based on the first convergence criterion. The termination of the criticality search cyclic modes are determined by a  $k_{eff}$  convergence sentinel rather than standard sentinel based on point number density errors. The outer iteration adjusts the cycle length or charge enrichment until the  $k_{eff}$  is close to the target value, as determined by the second convergence criterion. The first convergence criterion must be as tight or tighter than the second for the problem to converge.

## 2.2 Improvements to the Equilibrium Cycle Methodology

The REBUS-3 equilibrium cycle implementation discussed above does not allow microscopic cross sections to change with burnup, nor does it allow for burnup dependent fuel performance coupling. In some high-burnup reactors such as the TWR, the composition of the fuel will change dramatically through a in-reactor cycle. In these cases, the microscopic cross-sections will need to be updated at various burnup intervals to account for changes in spectrum and in energy self-shielding. Additionally, fuel behavior such as axial expansion, bond removal, and fission gas removal occur on long timescales in the equilibrium cycle. Traditional equilibrium searches do not account for these effects, all of which have been shown to have substantial effects on the equilibrium performance using explicit calculations [53]. In this section, we describe a from-scratch reimplement of the REBUS equilibrium cycle methodology that considers burnup-dependent microscopic cross sections and allows coupled fuel performance. The code is implemented within the framework of the TerraPower ARMI, a modular code system conceived of, created by, and supported by the author, designed to enable agile multiphysics and high-fidelity reactor analysis.

### 2.2.1 Description of the Advanced Reactor Modeling Interface

ARMI revolves around an object-oriented representation of a nuclear reactor, which is made up of assemblies, which are in turn made up of stacks of axial blocks. Each block contains components such as fuel, coolant, duct, cladding, wire wrap, etc., where each of these is assigned to a particular geometric size and material type affiliated with a material library. The material library contains mass fractions and temperature-dependent properties, including linear expansion coefficients, thermal conductivity, and heat capacity. BOL models are designed using a graphical user interface (GUI) and a simple spreadsheet. From an initial state, ARMI can produce inputs for a variety of solver kernels that either calculate an unknown at a particular snapshot in time (such as power density, neutron flux, temperature distributions, assembly bowing, cladding wastage, safety coefficients, etc.), or step through time to recalculate all such unknowns at a new timestep. When a solver kernel such as DIF3D [59] calculates flux and power, it stores the results as state variables in the ARMI model on the block level. At every timestep, ARMI transmits its current state to a database, which can be reloaded for later additional analysis or restarts.

With the reactor represented as a software object, users or modules of the code can easily modify the core. For instance, the reactivity coefficient module calls the voidCore method of a reactor to void all of the coolant. It then recalculates cross sections and flux to produce the coolant void coefficient. ARMI includes many input-generating and output-reading modules that allow external codes to be integrated into ARMI. In this way, ARMI allows a high level of automation to nuclear analysis. The major features of ARMI are summarized in Table 2.1.

### 2.2.1.1 ARMI Neutronics Capabilities

ARMI primarily relies on the MC<sup>2</sup>-2 [60] and DIF3D [59] codes for generating global flux and power. Depletion is accomplished either through REBUS-3 or through an internal matrix-exponential depletion solver. An internal replacement for DIF3D is also available, but is typically only used for special purposes discussed in Chapter III. At each cycle, MC<sup>2</sup>-2 inputs are generated and executed for representative fuel blocks of each burnup group and assembly type in parallel. A group of 180 fission products are lumped into 5 lumped fission products (LFPs), one for each parent nuclide amongst U-235, U-238 and Pu-239 through Pu-241. Once all microscopic cross section output data are available, they are merged into a single binary library in ISOTXS format [61]. ARMI then calls the neutronics kernel to generate the global multigroup flux. An assortment of other code modules then execute in sequence to determine additional state variables, followed by a depletion step. The loop continues until all cycles are computed.

For spatially-dependent reactivity coefficients, ARMI generates a sequence of DIF3D inputs, each with perturbations to coolant density, fuel density, structural density, and fuel temperature in various spatial regions of the core. It executes them in parallel, combining  $k_{eff}$  differences with the ARMI-computed delayed neutron fraction  $\beta_{eff}$  and the temperature-dependent instantaneous thermal expansion coefficients from the material library to produce reactivity coefficients. A perturbation-theory module has also been developed to more rapidly accomplish the same computation.

In addition, ARMI contains homogenized and heterogeneous input generators for

Table 2.1: Major Features of ARMI.

<ul style="list-style-type: none"> <li>• Arbitrary fuel management             <ul style="list-style-type: none"> <li>– Parallel branch searching</li> </ul> </li> <li>• Coupled thermal hydraulics             <ul style="list-style-type: none"> <li>– Automatic orifice zone generation</li> <li>– Hot density calculated in 3-D distribution</li> <li>– Material library with temperature-dependent properties</li> </ul> </li> <li>• Burnup-dependent microscopic cross sections             <ul style="list-style-type: none"> <li>– Classic and burnup-dependent LFP models</li> </ul> </li> <li>• Coupled fuel performance</li> <li>• Critical CR iteration</li> <li>• 3-D reactivity coefficient generation</li> </ul>	<ul style="list-style-type: none"> <li>• Graphical geometry input</li> <li>• Automatic density homogenization             <ul style="list-style-type: none"> <li>– Dimensions at cold shutdown, hot full power, etc. automatically generated</li> <li>– Heterogeneous and homogenized models available without extra effort</li> </ul> </li> <li>• SQL data persistence</li> <li>• Eulerian and Lagrangian history summaries</li> <li>• Economics assessments of fuel cycle costs</li> <li>• Full Python interface</li> </ul>
---	---

MCNPX [62]. Thus, given an ARMI input, an user can produce nominally identical DIF3D and MCNP files, with or without pin detail, allowing detailed code-to-code comparisons at a variety of conditions such as hot full power, (possible due to the automated thermal expansion), and robust calculations of control rod (CR) worth and shutdown margins. ARMI can produce additional input for the TerraPower proprietary code MCNPXT, which in turn extends MCNPX, allowing for fuel shuffling, coupled fuel performance, optimized depletion, and very large problems.

### 2.2.1.2 ARMI Thermal Hydraulics Capabilities

ARMI has several thermal hydraulics (T/H) kernels that produce temperature, flow, and pressure distributions throughout the core after power distributions have been produced by a neutronics kernel. Most simple is the internal module *thermo*, a 1-D solution of the energy balance and heat conduction equations through an average channel of each assembly. Thermo provides nominal temperatures and flow rates for scoping studies of the orifice settings. For more detailed analyses, the *subchan* module [63] produces non-communicative subchannel results considering 1-, 2-, and 3- $\sigma$  hot channel factors based on the Clinch River Breeder Reactor (CRBR). These modules are used to set assembly flow rates throughout the lifetime of a core so as to not violate materials-based temperature limits regarding thermal creep and fuel-cladding chemical interaction (FCCI). These flow rates in turn determine the pressure drop across each assembly, which is an important factor in determining how much each assembly distorts during its residence. In high-burnup reactors, all these considerations are particularly essential [64].

Coupled T/H is accomplished in ARMI with operator splitting. At the user's request, the temperature calculated for a given timestep will feed back into an updated neutronics input with thermal expansion (based on the material library) altering the number densities and fuel temperature (for Doppler broadening) in each fuel block. This new input is run, and the resulting power distribution replaces the first guess. Studies have shown that the effects of coupling T/H with steady-state neutronics in high-burnup fast reactors is small [53].

More detailed analysis can be done with the external code and SUPERENERGY

[65], as ARMI will write inputs for this, but it is not yet coupled back into ARMI. Interactions with COBRA [66] are under development.

### **2.2.1.3 ARMI Transient Capabilities**

ARMI has an interface to the SASSYS/SAS4A [67] code. The required reactivity coefficients, including the radial expansion coefficient, the 3-D fuel, coolant, and structure density coefficients, the Doppler coefficient, the voided Doppler coefficient, and the control rod worth curves, are all automatically computed by ARMI and appended as valid SASSYS input to a pre-made input file containing the rest of the plant model (primary system, secondary system, etc.). ARMI also automatically writes power distributions and core dimensions to the input. Finally, ARMI runs SASSYS, reads the output, and produces plots and reports based on the results for various design basis and beyond design basis transients. Statistical analysis options are also available for uncertainty propagation.

### **2.2.1.4 Implications of the ARMI code**

ARMI is a modular reactor suite that automates many essential aspects of reactor analysis. Given a single model, the entire life of the core may be simulated with transient analysis at key points of interest with little user interaction. It relies primarily on off-the-shelf simulation kernels without modifications, allowing users to take full advantage of the immeasurable effort (and pedigree) that the reactor analysis community has developed in the current and legacy codes. But it reanimates these codes such that the value of the collection is substantially greater than the sum

of the individual codes.

Besides routine core analysis, the ARMI architecture is conducive to research and optimization efforts, handling nuclear reactor data management in an elegant and transparent manner. This makes ARMI an attractive framework within which to develop Enhanced Equilibrium Cycle and, in Chapter V, to perform multidisciplinary design optimization.

### **2.2.2 Description of Fuel Performance Coupling Required for Accurate Equilibrium Cycle Calculations**

As any nuclear fuel is irradiated, on the order of 10% of the resulting fission products are gaseous xenon and krypton [68]. These gases internally pressurize the fuel, and in the case of metallic fuel cause dramatic swelling, with soft contact with the cladding occurring within the first few percent burnup. Early metallic fuel designs unsuccessfully tried to restrain this swelling by encapsulating it with stronger cladding, but the issue was eventually solved by leaving room for the fuel to swell by reducing the smear density [14]. Once gas bubbles account for about 30% of the fuel volume, they begin to interconnect and release the gas to the plenum. After the bubble interconnect, solid fission products continue to build up and cause swelling, but at a much slower rate. Eventually, the solid fission products fill the porous space, causing hard contact that will quickly cause cladding strain limits to be violated. Therefore, the onset of hard contact can be considered as a constraint for fuel-cladding mechanical interaction (FCMI).

Three neutronicly important phenomena occur as a result of the fission gas

evolution. Low smear density fuel pins typically contain a sodium bond between the cladding and the fuel to allow thermal transport through the fuel before it makes soft contact with the cladding. As the fuel swells, this sodium bond gets squeezed out of the pin and pushed into the gas plenum space. Sodium moderates the neutrons such that its removal hardens the spectrum and increases reactivity. Meanwhile, the fuel swells axially as well as radially, increasing the fuel length. The resulting decrease in fuel density increases neutron leakage, reducing reactivity. Finally, when the fission gas bubbles interconnect and release the gas, the highly-absorptive xenon escapes from the neutronic-important regions, thereby increasing reactivity. When the pores are large enough, the sodium bond in the plenum can infiltrate into the fuel, bringing sodium back into the system while increasing thermal conductivity.

Irradiation tests at EBR-II on a variety of metallic fuel alloys (U-Fissium, U-Zr, U-Pu-Zr) have produced enough data to develop mechanistic metallic fuel performance simulations such as the FEAST code [69] and the finite-element based ALCHEMY code (under development at TerraPower) that can predict these behaviors for various fuel designs immersed in various irradiation histories. ARMI interacts with both of these codes, thereby enabling coupled neutronics and fuel performance, which accounts for the three challenging-to-model but nonetheless essential phenomena.

### **2.2.3 The Burnup-Dependent Cross Section Model**

Fast reactor cross section codes such as MC<sup>2</sup>-2 [60] and ECCO [70] solve the neutron slowing down equation considering energy self-shielding, unresolved resonances,

and inelastic scattering to compute a detailed, composition-dependent neutron spectrum. The spectrum is used to collapse composition-independent fine-group microscopic cross section libraries into coarse-group libraries for use in global transport calculations. In typical fast reactor designs, composition changes due to burnup are slow relative to the residence time of the fuel, and the cross sections for this assembly are valid throughout. Constant cross sections through life were a requirement for early versions of REBUS, but applications in thermal research reactors motivated the development of cubic spline burnup interpolation for several MCNP-generated cross sections [71]. In high-burnup fast reactors, the composition changes can be substantial, such as the in-core conversion of depleted uranium to 12% Pu in the TWR. These reactors require cross sections that follow their composition changes throughout their life. Avoiding the need to prepare burnup dependent cross section libraries a priori, a dynamic burnup-dependent cross section model has been developed.

The burnup-dependent cross section model defines several initial types of assemblies at BOL, one for each initial enrichment or otherwise unique assembly design. At each depletion step, the burnup of each axial block of each assembly in the core is tracked in a table. A burnup group structure is defined as a list of  $N$  burnup upper bounds  $b_n$  in %FIMA, such that a fuel block is said to be in burnup group  $n$  if it has burnup  $b$  with  $b_{n-1} \leq b < b_n$ . When cross sections are generated, the block with the median burnup in each group for each assembly type is chosen from the table. In this way, the burnup group structure determines dictates the number of MC<sup>2</sup> - 2 cases that must be executed. A cross section set is generated for this particular block

that is then applied to all other blocks of the same type in burnup group  $n$ . The modular design of ARMI enables the cross sections to be computed with MC<sup>2</sup>-2 in parallel during a calculation, leading to a cross section library that increases in size during a simulation.

#### 2.2.4 Implementation of Enhanced Equilibrium Cycle

Enhanced Equilibrium Cycle is built as an ARMI module that uses any of the available neutronics kernels to compute the flux and the ARMI matrix-exponential depletion solver to compute transmutations and decay. The initial flux guess is produced by running a neutronics solution from the user-supplied BOL state. Contrary to the general REBUS formulation, EEC is implemented with no spatial averaging over the assemblies. This restricts users to purely physical fuel motions, but is not overly constrictive due to the high speed of current computers. Fuel motion is based on a location schedule, which is a list of core locations through which fuel will repetitively travel. The location schedule may be input directly for complicated fuel management schemes, or may be generated from a simple ring schedule list, allowing the user to specify the general direction of fuel as it moves from its initial charge, through the reactor, to its discharge. The user may therefore specify convergent or divergent shuffling, or any hybrid combination of these with the ring schedule. For example, in a 16-ring core, convergent shuffling would be specified with a ring schedule of [16, 1] where the resulting fuel management would charge assemblies at ring 16, converge them ring-by-ring until finally they discharge from ring 1. Alternatively, assemblies with a ring schedule of [16, 9, 1, 8] would charge at ring 16, converge to

ring 9, then jump to ring 1 and move outwards to ring 8, finally discharging from ring 8. With ring schedules, fuel moves in spirals from ring to ring, skipping the number of locations equal to the number of discharges per cycle. This means every assembly is moved at every cycle. Long durations of fuel management operations in pool-type SFRs may prohibit this magnitude of fuel motion in large cores, inspiring more efficient fuel management. However, by delaying the motion of assemblies rather than moving them through their location schedules, these ideal equilibrium shuffling routines can be largely replaced by more efficient shuffling without changing the equilibrium performance. As is demonstrated, the amount of input to set up a wide variety of fuel management plans is small and, most importantly for design optimization, simple to automate.

Given the initial flux and cycle length, EEC generates a first guess of the equilibrium state by performing the first cyclic mode iteration, solving Eq. (2.15) according to the location schedule. The number of independent cascades to be depleted is determined by the user's requested number of assemblies to discharge per cycle. The composition of the freshest assembly in each cascade is set to that of the user-supplied charge assembly. This fresh assembly is depleted with the current flux guess in its region for a full burn cycle, and the resulting EOC densities are set as the BOC composition of the next assembly. The process repeats until the discharge density for each cascade is computed. With updated number densities, a region-density iteration begins with burn matrix averaging and extrapolation to determine the cycle-averaged  $B$  matrices as discussed in §2.1.3.2. In all cyclic-mode iterations besides the first, the cycle-averaged  $B$  matrix is used in the solution of Eq. (2.15), such that the

neutronic computations only occur within the region-density iteration. Next, fuel performance coupling updates the fission gas removal, bond squeeze-out, and axial expansion throughout the reactor. With the new core compositions determined, the burnup-dependent microscopic cross sections are then recalculated. Finally, DIF3D is called to update the flux distribution and the iteration continues. This enhanced scheme is shown in Figure 2.4.

Two levels of parallelism are possible in EEC. In the solution of Eq. (2.15), each cascade is independent of the next. These calculations are therefore done in parallel. The speedup due to parallelism achievable at this level is equal to the number of assemblies discharging per cycle, up to the number of processors available. If, for instance, there were 12 discharging assemblies per cycle and 16 available processors, the first 12 processors would each deplete  $\frac{1}{12}$  of the fuel blocks in the core, and the remaining 4 processors would remain dormant for a total speedup factor of 12. On the other hand, the depletion calculations in the region-density iterations are all independent. Here, each processor in the example would deplete  $\frac{1}{16}$  of the fuel blocks, with none left dormant. This typically allows for a larger speedup, because there are usually fewer discharging assemblies than available computer cores. Parallelism is coded using MPI within ARMI.

For this research, EEC has been implemented for the in-core portion of the equilibrium cycle methodology only. The external cycle capabilities of REBUS do not require any modification to work with the high-burnup reactors of interest.

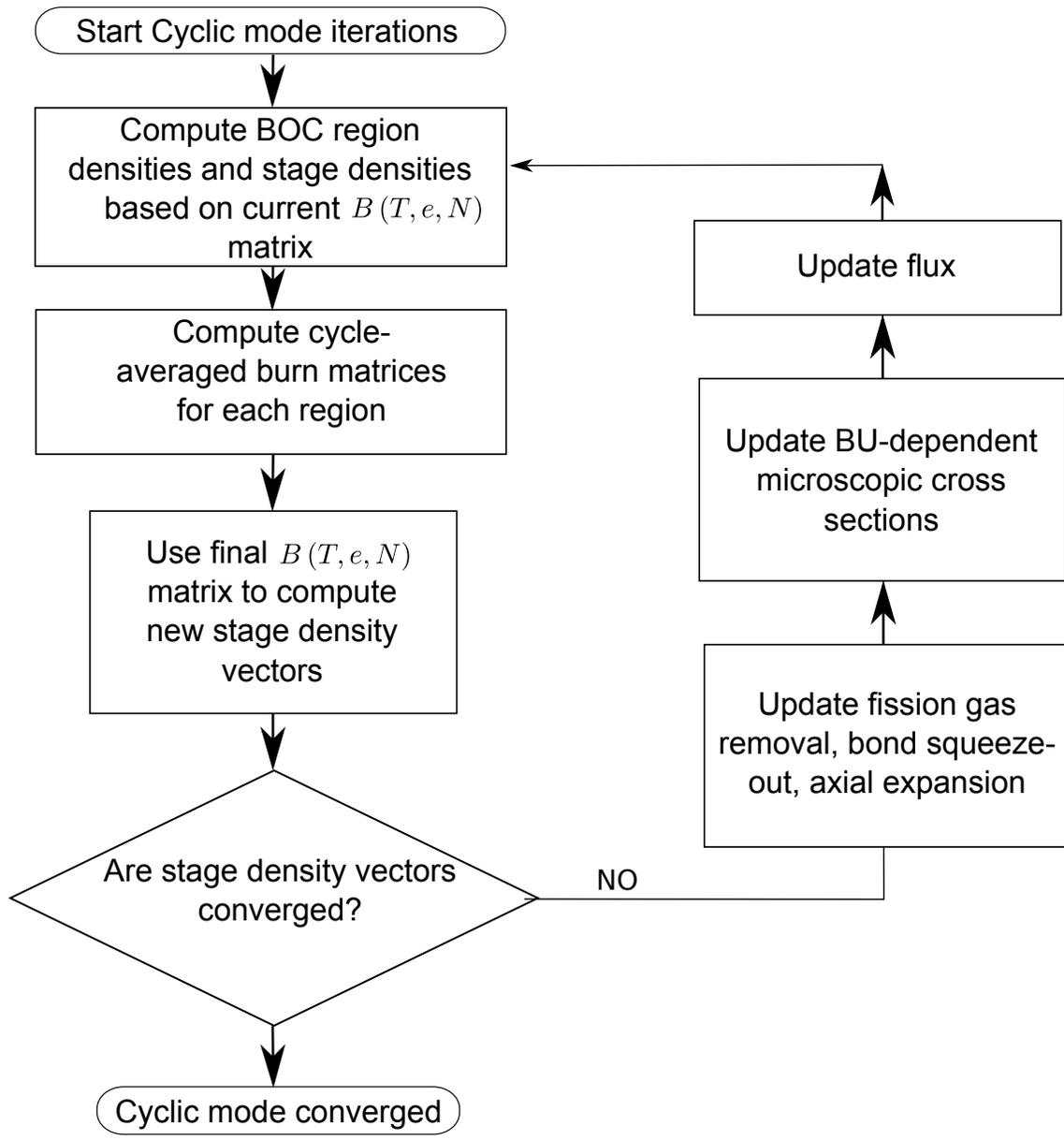


Figure 2.4: The Enhanced Equilibrium Cycle flow chart.

## 2.3 Equilibrium Calculations and Sensitivities with the Enhanced Equilibrium Cycle Methodology

Results from sample calculations using EEC on a TWR-like reference core are discussed in this section.

### 2.3.1 Description of the Reference Core

A reference TWR core is used for all sample calculations. It is a 1200 MWt SFR operating in traveling wave mode, meaning it sustains criticality indefinitely given only depleted uranium without requiring further enrichment or separations. Fuel management is purely convergent. The assembly design (summarized in Table 2.2) is not expected to reach the burnup and fluence requirements of a TWR due to fuel performance issues and perhaps duct distortion limits, but is adequate for understanding the behavior and success of EEC.

There are 10 control rods and 3 diverse safety rods with natural or  $B_4C$  poison. Rods are fully withdrawn unless otherwise noted. The core is modeled in 1/3 symmetry with the DIF3D hexagonal-nodal diffusion theory, converging with  $\epsilon_k = 10^{-7}$ . Cross sections are generated in the standard constant lethargy 33-group energy structure.

Table 2.2: Reference TWR core design parameters.

Item	Value
Power	1200 MWt
Smear density	75%
Duct thickness	3.0 mm
Cladding thickness	0.45 mm
Initial fuel O.D.	8.66 mm
Number of pins/assembly	169
Fuel	U-10Zr w/ Na-bond
Feed U-235 enrichment	0.3 wt%
Structural material	HT-9
Assembly pitch	167.5 mm
Duct-to-duct gap	15 mm
Wire wrap O.D.	1 mm
Fueled height	2.5 m
Core Inventory	147 MT HM
Number of assemblies	517
Cycle length	550 days

Table 2.3: Comparison of the  $k_{eff}$  and discharge burnup of the direct and equilibrium mode cases.

	Direct Method	Equilibrium Method
BOEC $k_{eff}$	1.00990	1.00993
Discharge Burnup (% FIMA)	36.41	36.51

### 2.3.2 Investigation of the Equilibrium Search Implementation and Burn Matrix Averaging

The first set of cases are designed to demonstrate that the fast equilibrium search methodology works as implemented and to quantify the effect of performing cycle-averaging of the burn matrix  $B$ . All cases in this section use an identical and stationary microscopic cross section set generated from approximate equilibrium number densities.

#### 2.3.2.1 Direct Shuffling vs. Equilibrium Search

To verify the implementation of the equilibrium methodology, a case where fuel was explicitly shuffled according to the location schedule is compared to an equilibrium-mode calculation. Neither case included a region-density iteration, but both had 2 burn steps, each of length 275 days. The results in Table 2.3 show that the eigenvalue from the two methods agree within 4 pcm at BOEC and that the discharge burnups are within 0.1% FIMA of each other. This level of agreement confirms the accuracy of the implementation of the equilibrium search.

To compare the convergence of the two methods, the reactivity difference between subsequent  $k_{eff}$  values of both cases are shown in Figure 2.6. Each point represents a cyclic iteration, each requiring roughly the same amount of computation for both

cases. The true convergence criterion for both cases required that the relative difference between any two number densities in any fuel block between iterations is less than 0.001. Based on number densities, the direct case converges in 95 cycles while the equilibrium case converges in 26. Oscillations in  $k_{eff}$  for the converged direct case are on the order of 0.2 pcm, which they are much more tightly converged for the equilibrium case, with oscillations on the order of 0.001 pcm. It is worth noting that the equilibrium case attained  $k_{eff}$  convergence  $< 1$  pcm after just 10 iterations even though final convergence did not occur until after 26 iterations. This level of convergence, which is not obtained in the direct case until after 60 iterations, may be adequate for many applications, suggesting that the equilibrium mode is a factor of 6 faster. Considering that direct calculations can take on the order of 10 hours, this factor is very substantial for a design team.

### 2.3.2.2 Burn Matrix Averaging and Burn Step Length

As nuclides transmute, the reaction rates change, altering the fission rate, among others. To maintain constant power operation, the flux magnitude in the reactor must change to counter any change in fission rate such that  $\phi \Sigma_f = \text{constant}$ . The flux magnitude affects the rate of change of  $\Sigma$ , and the problem of flux normalization is therefore nonlinear. One must either use short enough timesteps at which to recalculate/renormalize the flux or use average burn matrices for each timestep. The region-density iteration of §2.1.3.2 is designed to produce cycle-averaged burn matrices for this purpose. To quantify the effect of these iterations, cases were executed that do and do not average the burn matrices. Additionally, cases with very

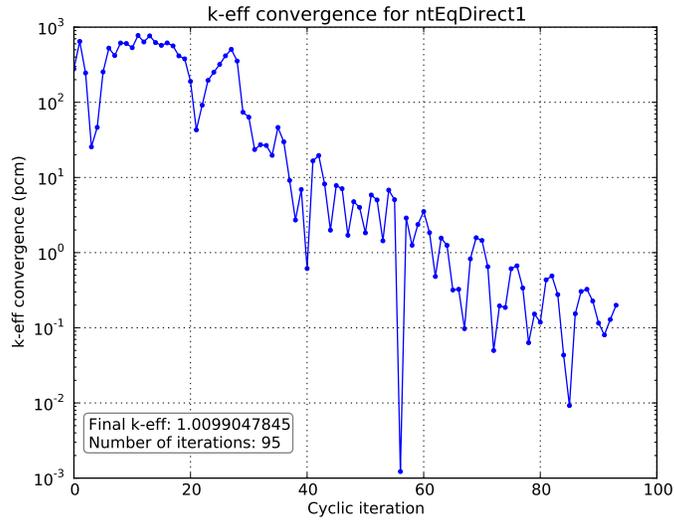


Figure 2.5: Convergence of a direct equilibrium cycle calculation in terms of  $k_{eff}$ . The value oscillates around the final value and was randomly very close to it around iteration 55, but this should not be confused for a converged value. Recall that the convergence criteria in this case is driven by the region number densities. The eigenvalue convergence for each cyclic iteration was such that all digits are relevant.

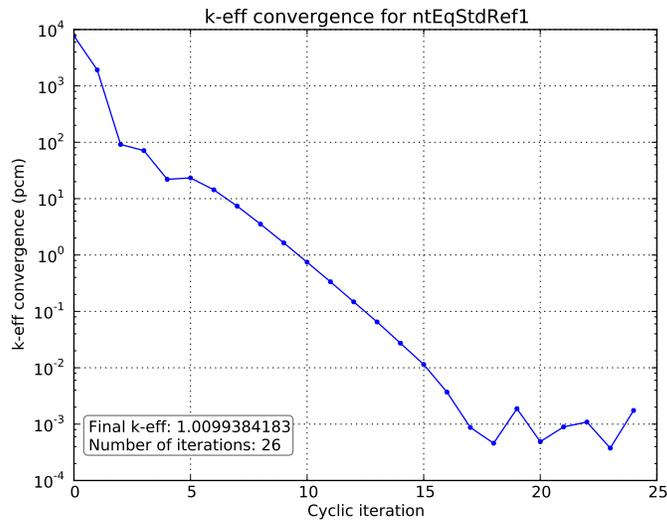


Figure 2.6: Convergence of an equilibrium-mode calculation in terms of  $k_{eff}$ .

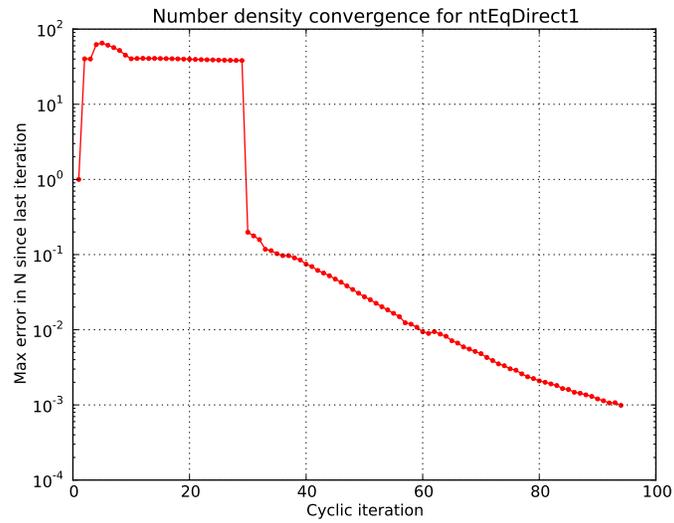


Figure 2.7: Convergence in terms of peak number density error between cyclic iterations in a direct equilibrium calculation. The sharp drop represents the iteration where all the fresh fuel had discharged from the core.

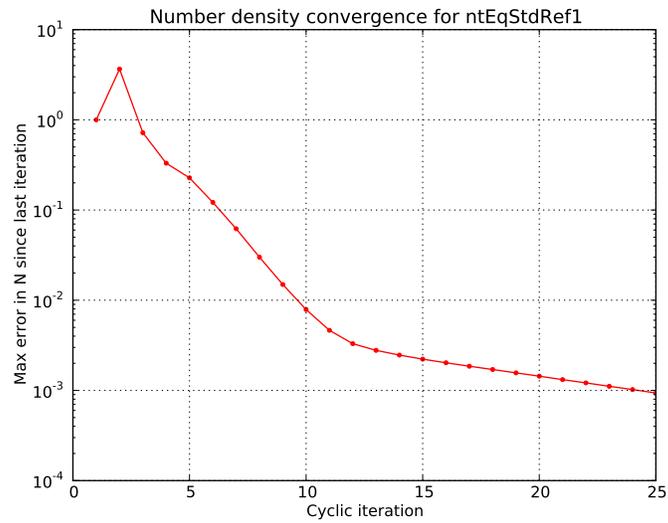


Figure 2.8: Convergence in terms of peak number density error between cyclic iterations in a fast equilibrium-mode calculation.

many burn steps were executed both with and without averaging. The results shown in Table 2.4 show that in the reference core investigated, burn matrix averaging is worth up to 80 pcm in BOEC  $k_{eff}$ . Burn matrix averaging requires another level of iteration within each cyclic iteration and therefore slows down the calculation. For scoping studies in high-burnup fast reactors, there is low motivation for performing burn matrix averaging.

Table 2.4: The effects of burn matrix averaging and burn step length. The reference case is the Direct2 case, with  $k_{eff} = 1.010717$ .

Case Title	Averaging	Type	Burn Steps	$\Delta k_{eff}$ (pcm)	Cyclic Iterations
StdRef1	N	EQ	2	-77.0	26
StdRef2	Y	EQ	2	23.6	26
HiFi1	N	EQ	10	3.0	26
HiFi2	Y	EQ	10	22.6	26
Direct1	N	D	2	-80.3	95
Direct2	N	D	10	0.0	95

56

Table 2.5: The effects of burnup dependent microscopic cross sections on the equilibrium cycle methodology.

Case Title	BU Groups	BU Structure	$\Delta k_{eff}$ (pcm)	Cyclic Iters.
Reference	4	[2,5,15,100]	-2	25
Stationary	4	[2,5,15,100]	-11	24
10 Groups	10	[2,5,8,11,14,17,20,23,26,33,100]	-	26
1 Group	1	[100]	-501	28
2 Groups	2	[2,100]	17	26
3 Groups	3	[2,5,100]	10	27
5 Groups Low	5	[1,2,5,15,100]	-6	25
5 Groups High	5	[2,5,15,30,100]	-2	25

### 2.3.3 The Effects of Burnup Dependent Cross Sections on the Equilibrium Search

As discussed in §2.2.3, cross sections in high-burnup fast reactors change with burnup. Several cases were chosen to demonstrate the dynamics of burnup dependent cross sections, with results summarized in Table 2.5, where  $k_{eff}$  is compared to the highest-fidelity case with 10 burnup groups, and in Figure 2.9 showing absolute values of  $k_{eff}$ . Each case used burn matrix averaging. With a single burnup group, a 500 pcm error was found in BOEC  $k_{eff}$ , demonstrating the large error made by assuming constant 33-energy group cross sections. With just two burnup groups, the error is only 17 pcm, with marginal improvement when additional groups are chosen. A simple 2-group model is sufficient to capture the reactivity effects in high burnup fast reactor calculations, but a 1-group model is unacceptable.

### 2.3.4 The Effects of Coupled Fuel Performance on the Equilibrium Search

As discussed in §2.2.2, fuel performance coupling is expected to have substantial effects on equilibrium studies. Results from fuel performance coupling cases are shown as a collection of five cases in Table 2.6 that individually resolve the effects on the equilibrium state in terms of the three primary changes that evolve with burnup. The same results shown in Figure 2.10 show the convergence of  $k_{eff}$  as the equilibrium search proceeds. Seen graphically, it is clear that neglecting all of the fuel performance phenomena results in substantial cancellation of error, and may be acceptable for scoping studies. More sophisticated multiphysics calculations require the rigor provided by the advanced modeling. For example, the correct

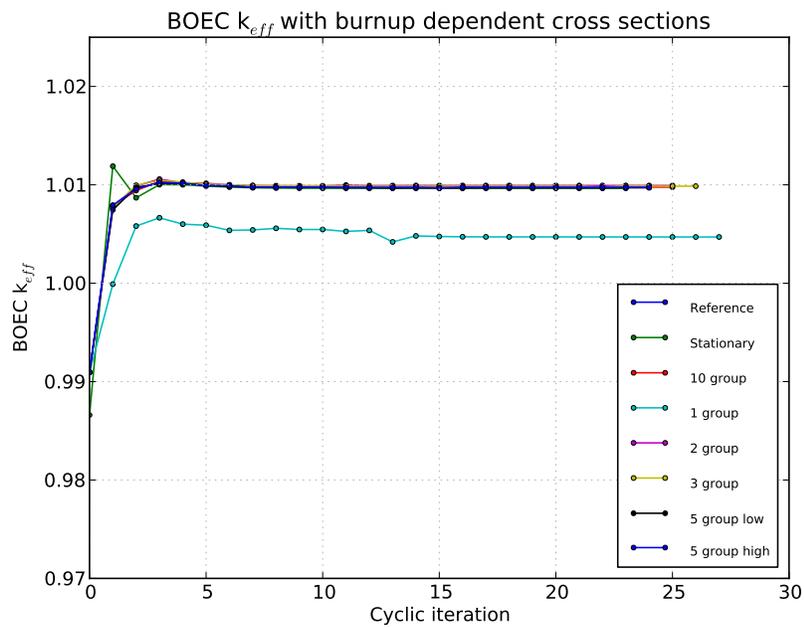


Figure 2.9: Effects of burnup dependent cross sections on BOEC  $k_{eff}$ .

Table 2.6: The effects of coupled fuel performance, including axial expansion, fission gas removal, and bond squeeze-out on the equilibrium cycle methodology.  $\Delta k_{eff}$  is compared to the highest-fidelity case treating all effects.

Case	Discharge BU	$\Delta k_{eff}$ (pcm from ref.)	Cyclic Iterations
All effects	36.2	-	46
Axial expansion	36.6	-816	45
Bond squeeze-out	35.7	1165	29
Fission gas removal	36.1	355	24
No effects	36.1	225	24

sodium inventory must be known to accurately compute the sodium density reactivity coefficient. The coefficient is proportional to the sodium inventory in the assembly, including that in the bond.

The additional state perturbations inflicted by the fuel performance coupling slow the convergence of the equilibrium cycle substantially. In particular, the noise from the axial expansion coupling nearly doubles the number of cyclic iterations that must be computed to reach a certain convergence criterion.

### 2.3.5 The High Burnup Equilibrium Instability

Simulations that reach sufficiently high peak burnup in central regions of the core have been found to exhibit non-convergent oscillatory behavior. Reactivity typically increases with burnup for low- to mid-burnup cases, as Pu-239 and other actinides build to their equilibrium levels. Burnup beyond a spectrum-dependent threshold (around 40% FIMA in a metallic fueled SFR) leads to increasing concentrations of fission product poisons that overpower the reactivity of the actinides. In equilibrium calculations, the composition of any region can vary dramatically between cyclic iterations, and has been found to do so in such high burnup situations. Double-

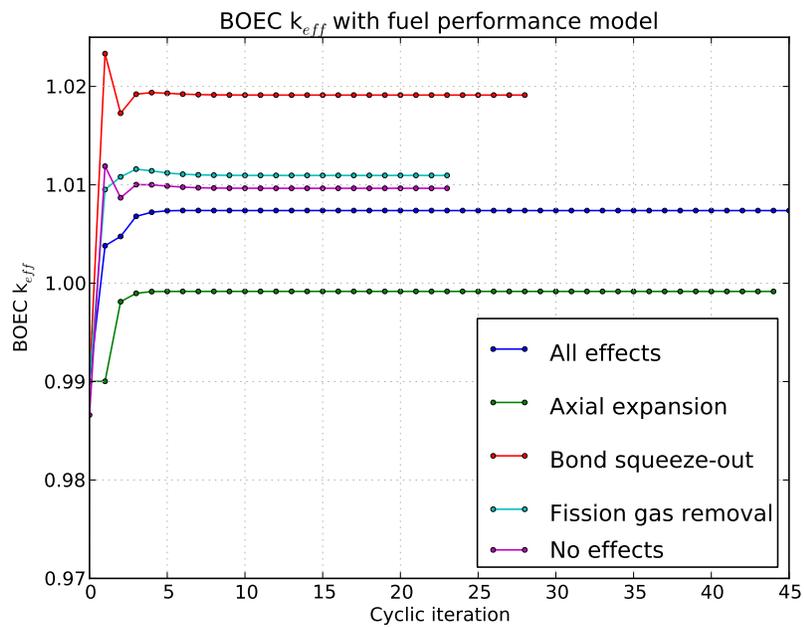


Figure 2.10: Effects of fuel performance feedback on BOEC  $k_{eff}$ .

hump cases in which the central region produces less power than the top or bottom regions result in an axial shift in power towards the top or bottom. In the following cyclic iteration, the entire core will deplete according to this axial shift. This positive feedback eventually pushes the majority of the power towards the top or bottom. Each subsequent cyclic step depletes the powered region beyond the burnup at which the fuel is reactive, and as a result the following flux update computes most of the power as residing in the opposite region. The power and burnup then jump from top to bottom at each iteration and the case fails to converge, as seen in Figure 2.11. The instability has been found to be independent of the number of subintervals considered within a cyclic mode iteration.

Conversely, in explicit calculations, each depletion step can change the composition of a region by only a small amount from its previous composition. As a result, asymmetries in flux that cause the instability in equilibrium cases self-correct on a cycle-by-cycle scale. A small (single depletion step) asymmetric burn towards the top of the core is followed by an equally small correction towards the bottom and the case remains convergent. The self-correction timescale is on the order of the depletion step length. Thus, equilibrium cases suffer from the fact that their depletion steps are the entire residence time of each fuel assembly rather than just a single cycle.

The high burnup equilibrium instability represents an inability for equilibrium mode calculations to model asymmetrically-reactive models. For practical designs, the neutronic threshold burnup is beyond the fuel performance-limited burnup limit, and this limitation does not represent a substantial flaw. However, if advanced fuels

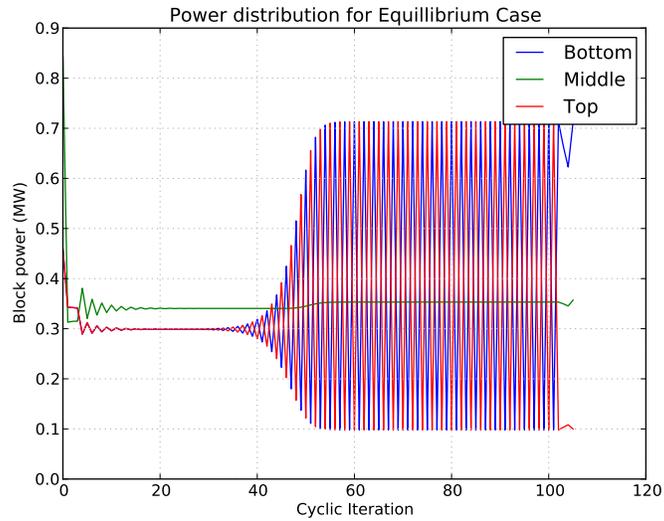


Figure 2.11: Power of a single assembly in an equilibrium case demonstrating the high burnup equilibrium instability.

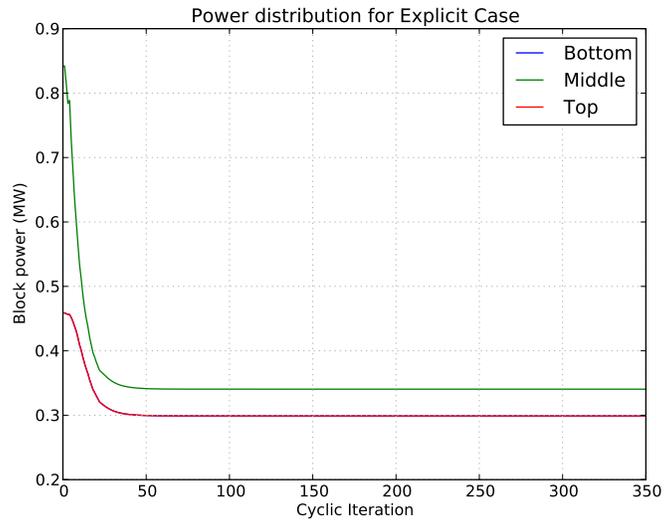


Figure 2.12: Power of a single assembly in an explicitly shuffled equilibrium case demonstrating the stability of the explicit method.

that can enter such deep-burn regimes are created in the future, this issue could become more relevant.

## CHAPTER III

# Modal Expansion Perturbation Theory

While the Enhanced Equilibrium Cycle method developed in Chapter II provides rapid solutions of the equilibrium state of a reactor, its speed is not sufficient to allow practical design optimization. Each design change must be evaluated through the equilibrium method, requiring several full neutronics solutions to be calculated. Inspired by the desire to accelerate equilibrium cycle calculations for design changes around a reference state, we develop in this chapter a Modal Expansion Perturbation Theory (MEPT) which forms an algebraic expression that may be evaluated in lieu of neutronics solutions to approximate the effects of design perturbations on the flux and power shapes. MEPT acts as a perturbation method at a snapshot in time which will be applied to the equilibrium cycle in Chapter IV.

### 3.1 Derivation of Modal Expansion Perturbation Theory

Before deriving the modal expansion perturbation theory, we review the derivation of the standard neutron diffusion equation, based on the DIF3D finite-difference

formulation [59]. This particular approximation of the transport equation is used as a convenient framework within which to develop the theory, but does not preclude the applicability of the theory in other approximations.

### 3.1.1 The Finite-Difference Solution of the Steady-State Neutron Diffusion Equation

The group balance statement for the steady-state multigroup neutron diffusion equation can be written in matrix notation as

$$(D_g + \Sigma_g) \phi_g - \sum_{g' \neq g} T_{gg'} \phi_{g'} = \frac{1}{\lambda} \chi_g \sum_{g'=1}^G F_{g'} \phi_{g'}, \quad (3.1)$$

where  $\phi_g$  is the  $N$ -dimension scalar neutron flux vector on the finite-difference mesh, and  $N$  is the number of cells in the spatial mesh. The matrix  $D_g$  represents the leakage of neutrons out of energy group  $g$  by diffusion,  $\Sigma_g$  represents the removal of neutrons from group  $g$  by absorptions or out-scattering,  $T_{gg'}$  represents in-scattering (and the n,2n scatter source) from group  $g'$  into  $g$ ,  $F_{g'}$  represents the number of neutrons produced by fissions caused by neutrons in group  $g'$ , and  $\chi_g$  represents the fraction of fission neutrons that are born with energies in group  $g$ . These  $N \times N$  matrices are each diagonal and are defined as:

$$\Sigma_g = \text{diag}(\Sigma_{r,g} V_l), \quad (3.2)$$

$$T_{gg'} = \text{diag}(\Sigma_{s,gg'} V_l), \quad (3.3)$$

$$F_g = \text{diag}(\nu\Sigma_{f,g}V_l), \quad (3.4)$$

and

$$\chi_g = \text{diag}(\chi_g), \quad (3.5)$$

where the subscript  $l$  represents each spatial region and  $V_l$  is the corresponding volume of region  $l$ .

Combining the balance statement for each energy group  $g$ , the multigroup equations may be written as a single generalized eigenvalue matrix equation:

$$M\phi = \frac{1}{\lambda}B\phi, \quad (3.6)$$

where  $M$  and  $B$  are square matrices of the order  $N^*G$ ,  $G$  is the number of energy groups, and  $\lambda$  is the fundamental eigenvalue (also known as  $k_{eff}$ ). Note that the symbol  $B$  was also used for the burn matrix in Chapter II, but the distinction should be obvious from context. The matrix  $M$  represents all the loss terms, as well as the transfer of neutrons between energy groups by scattering, and is given by:

$$M = \begin{bmatrix} [A_1] & & & & [0] \\ & [A_2] & & & \\ & & \ddots & & \\ & & & & [A_G] \\ [0] & & & & \end{bmatrix} - \begin{bmatrix} [0] & [T_{12}] & \cdots & \cdots & [T_{1G}] \\ [T_{21}] & [0] & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & [T_{G-1,G}] \\ [T_{G1}] & [T_{G2}] & \cdots & [T_{G,G-1}] & [0] \end{bmatrix}, \quad (3.7)$$



$$[A_{gy}^{j,k}] = \text{diag}(-a_i^y)_{jkg}, \quad (3.10)$$

$$[A_{gz}^{j,k}] = \text{diag}(-a_i^z)_{jkg}. \quad (3.11)$$

Expressions for the  $a$  and  $b$  terms are geometry dependent, and are summarized in detail for Cartesian, triangular, and  $R - \theta$  geometries in the DIF3D manual [59].

The full, 3-D  $A_g$  matrix is 7-banded, with the central stripe representing absorptions and outscatters, the adjacent stripes representing leakage in the i-direction, the next two stripes on either side representing leakage in the j-direction, and the final two stripes representing leakage in the k-direction. A plot of an  $A_g$  matrix for energy group 4 from a small sample problem is shown in Figure 3.1. In 1/3 triangular geometry, the mesh numbers do not line up perfectly in all directions as they would in Cartesian geometry. As a result, the i- and j- direction leakage bands converge on the central diagonal as mesh indices that are not in the problem are encountered.

The matrix  $B$  represents neutron gains from fission and can be expressed as:

$$B = \chi F^T = \begin{bmatrix} \chi_1 F_1 & \chi_1 F_2 & \cdots & \chi_1 F_G \\ \chi_2 F_1 & \ddots & & \\ \vdots & & \ddots & \\ \chi_G F_1 & & & \chi_G F_G \end{bmatrix}, \quad (3.12)$$

where the  $N^*G$  by  $N$  matrices  $F$  and  $\chi$  are defined as

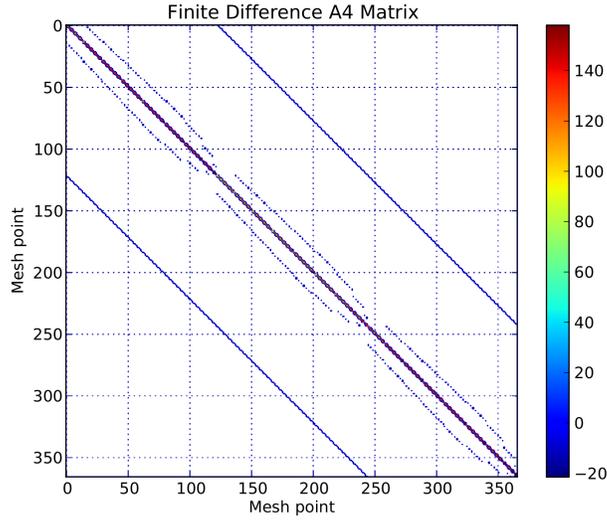


Figure 3.1: The  $A_g$  matrix of a 3-D finite-difference mesh of a small 1/3-symmetric triangular sample problem with three z-direction mesh points.

$$F = \text{col} [F_1, F_2, \dots, F_G], \quad (3.13)$$

and

$$\chi = \text{col} [\chi_1, \chi_2, \dots, \chi_G]. \quad (3.14)$$

$B$  is of order  $N \times G$  but has a rank of at most  $N$  because it is formed by an outer product and  $N$  is the maximum number of fission source points in  $F_g$ . In realistic reactor problems, mesh points that do not contain fission source material exist as reflectors, control rods, and other structures, so the rank of  $B$  is less than  $N$ .

The diffusion equation is a generalized eigenvalue problem which can be cast as

a standard eigenvalue problem by inverting  $M$  to write:

$$\lambda\phi = M^{-1}B\phi. \quad (3.15)$$

Equation (3.15) may be cast in operator form by defining  $L = (M^{-1}B - \lambda I)$  where  $I$  is the identity matrix, or equivalently,  $L = (M - \frac{1}{\lambda}B)$  to write:

$$L\phi = 0. \quad (3.16)$$

When solving these equations for the neutron population, inverting the large matrix  $M$  explicitly is impractical, as any direct method would require far too much memory. But because of its well-known and very sparse structure, the effective inversion of  $M$  can be accomplished iteratively and efficiently. Furthermore, exploiting the fact that the fission source matrix  $B$  is of rank  $N$ , practical diffusion solvers condense the multigroup problem from one of order  $N * G$  to one of order  $N$  by introducing a fission source vector of length  $N$ , called  $\psi$ . Rather than inverting  $M$ , the condensed problem requires implicitly inverting an order- $N$  matrix  $A_g$  for each energy group, further reducing the memory requirements of the algorithm. To condense the problem in this way, the fission source vector  $\psi$  is defined as:

$$\psi = F^T\phi = \sum_{g=1}^G F_g\phi_g. \quad (3.17)$$

The corresponding  $(N \times N)$  within-group balance equations from Eq. (3.1) become:

$$A_g \phi_g - \sum_{g' \neq g} T_{gg'} \phi_{g'} = \frac{1}{\lambda} \chi_g \psi. \quad (3.18)$$

Introducing the  $(N * G \times N)$  matrix  $C$  as:

$$C = \text{col} \left( \begin{bmatrix} C_1 \end{bmatrix}, \begin{bmatrix} C_2 \end{bmatrix}, \dots, \begin{bmatrix} C_G \end{bmatrix} \right) = M^{-1} \chi, \quad (3.19)$$

allows the full balance statement  $M\phi = \frac{1}{\lambda} \chi \psi$  to be cast as a mapping from the fission vector to the full  $(N \times G)$  flux vector:

$$\phi = \frac{1}{\lambda} C \psi. \quad (3.20)$$

An expression for  $C$  may be obtained by noting that for downscattering problems, the system  $MC = \chi$  is block lower triangular, and can be written out as:

$$\begin{aligned} A_1 C_1 &= \chi_1 \\ -T_{21} C_1 + A_2 C_2 &= \chi_2 \\ -T_{31} C_1 - T_{32} C_2 + A_3 C_3 &= \chi_3, \end{aligned}$$

which can be readily solved with block back-substitution to obtain each  $(N \times N)$  matrix  $C_g$ :

$$C_g = A_g^{-1} \left( \chi_g + \sum_{g' < g} T_{gg'} C_{g'} \right). \quad (3.21)$$

Implicitly solving Eq. (3.21) comprises the inner iterations of the diffusion solution.

Premultiplying the full eigenvalue problem in Eq. (3.15) by  $F^T$  gives:

$$\lambda F^T \phi = F^T M^{-1} \chi F^T \phi = F^T M^{-1} \chi \psi, \quad (3.22)$$

which, with substitutions, becomes the condensed  $N \times N$  standard eigenvalue problem:

$$Q\psi = \lambda\psi, \quad (3.23)$$

where we have defined:

$$Q \equiv F^T C = \sum_{g=1}^G F_g C_g. \quad (3.24)$$

The structure of  $Q$  is not sparse, and must not be formed explicitly. For a small sample problem, it is possible to form the matrix for study. Its structure is shown in Figure 3.2.

Equation (3.23) can be solved for the fission source using the same eigenvalue solver used to solve Eq. (3.15). The group fluxes  $\phi_g$  can be extracted from the fission source  $\psi$  with partial forms of Eq. (3.20):

$$\phi_g = \frac{1}{\lambda} C_g \psi. \quad (3.25)$$

The eigenvalue problem posed by Eq. (3.23) is traditionally solved with the power iteration, since the dominant eigenvector is of primary interest in most applications.

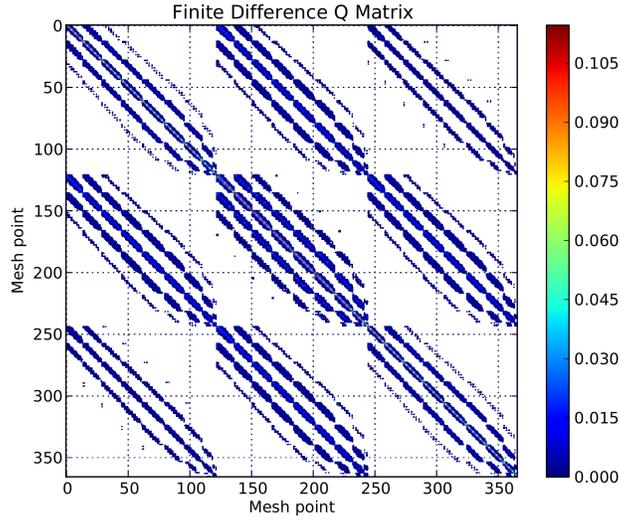


Figure 3.2: The  $Q$  matrix structure from a small sample problem.

Given an initial guess of  $\psi^0$ , subsequent eigenvector iterates are determined as:

$$\psi^n = \frac{1}{\lambda^{n-1}} Q \psi^{n-1}, \quad (3.26)$$

and

$$\lambda^n = \lambda^{n-1} \frac{\|\psi^n\|_1}{\|\psi^{n-1}\|_1}. \quad (3.27)$$

The repeated application of  $Q$  to the guess  $\psi^0$  followed by normalization allows the largest eigenpair to dominate all others and emerge. This process is known as the outer iteration. Because the second largest eigenvalue is that last one that must be eliminated to expose the dominant eigenpair, the rate of convergence of the power method is inversely proportional to the ratio of the second largest eigenvalue of  $Q$  to the largest, known as the dominance ratio. At each outer iteration in practical

reactor problems, the action of  $Q\psi^{n-1}$  is determined through Eqs. (3.24) and (3.25) by writing

$$Q\psi^{n-1} = \sum_{g=1}^G F_g C_g \psi^{n-1} = \lambda^{n-1} \sum_{g=1}^G F_g \phi_g^n, \quad (3.28)$$

where we have defined the current flux vector based on the previous fission source iterate

$$\phi_g^n = \frac{1}{\lambda^{n-1}} C_g \psi^{n-1}. \quad (3.29)$$

Like  $Q$ , the matrices  $C_g$  defined by Eq. (3.21) are not sparse and should not be formed explicitly. Instead, we multiply  $\psi$  to both sides of the matrix equation in Eq. (3.21) to obtain a linear system:

$$\begin{aligned} A_g C_g \psi &= \chi_g \psi + \sum_{g' < g} T_{gg'} C_{g'} \psi \\ A_g \phi_g^n &= \frac{1}{\lambda^{n-1}} \chi_g \psi^{n-1} + \sum_{g' < g} T_{gg'} \phi_{g'}^n. \end{aligned} \quad (3.30)$$

Solving Eq. (3.30) with an iterative solver for each group during each outer iteration allows the power method to proceed without forming any dense  $N \times N$  matrices.

By using the fission source problem, we reduce the task of inverting the ( $N * G \times N * G$ ) matrix  $M$  before solving the eigenvalue problem to one of inverting ( $N \times N$ ) matrices  $A_g$  a total of  $G$  times. Within this finite-difference neutron diffusion framework, we derive modal expansion perturbation theory to determine the flux

shape changes of perturbations on a reference solution to these equations.

### 3.1.2 Determining an Algebraic Expression of Perturbed Flux for Arbitrary Perturbations

Given a reference state of a reactor as represented by Eq. (3.16), we seek to determine the change in flux  $\delta\phi$  that arises due to a perturbation in the governing system  $\delta L$ . The perturbed system can be written as:

$$L'\phi' = 0, \tag{3.31}$$

where  $L' = L + \delta L$  and  $\phi' = \phi + \delta\phi$ . Expanding the perturbed system and truncating the second order term leads to an approximate statement relating the state perturbations to the desired flux perturbation:

$$\begin{aligned} (L + \delta L)(\phi + \delta\phi) &= 0 \\ L\phi + \delta L\phi + L\delta\phi + \delta L\delta\phi &= 0 \\ L\delta\phi + \delta L\phi &\approx 0. \end{aligned} \tag{3.32}$$

Directly solving for  $\delta\phi$  requires computations on the order of complexity required to solve the original reference system. This approach is therefore unfavorable for evaluating very many perturbations of detailed reactor models in design optimization. Rather than solving the system of equations, we seek algebraic expressions for the perturbed flux to maintain favorable computational efficiency.

The change in flux can be expanded on the characteristic basis of  $\lambda$ -eigenmodes of the unperturbed system. With the harmonic eigenmodes of this system written as  $\phi_i$ , the perturbed flux is expressed by the expansion:

$$\delta\phi = \sum_{i=0}^{\infty} a_i \phi_i. \quad (3.33)$$

Given pre-computed eigenmodes of the reference solution  $\phi_i$ , the expansion coefficients  $a_n$  are the remaining unknowns needed to approximate the perturbed flux. Expressions for these coefficients as functions of arbitrary design perturbations  $\delta L$  may be obtained using properties of the harmonics of the associated adjoint reference solution, where  $L^*\phi^* = 0$ , and  $L^*$  is defined as the solution to  $\langle\phi^*, L\phi\rangle = \langle L^*\phi^*, \phi\rangle$  for any  $\phi^*$ , and where  $\langle\cdot\rangle$  represents an inner product or integration over space and energy. Writing the eigenvalue problem in terms of all its harmonic solutions as  $L_i\phi_i = \left(M - \frac{1}{\lambda_i}B\right)\phi_i$  or  $(M^{-1}B - \lambda_i I)\phi_i = 0$ , where  $I$  is the identity matrix. We define the eigenvalue-invariant operator  $\hat{L} = M^{-1}B$ , allowing the eigenvalue problem to be written in standard form as

$$\hat{L}\phi_i = \lambda_i\phi_i. \quad (3.34)$$

Given an eigenvalue problem of Eq. (3.34) and the associated adjoint problem,  $\hat{L}^*\phi_j^* = \eta_j\phi_j^*$ , it is well known that the definition of the adjoint implies biorthogonality of  $\phi_i$  and  $\phi_j^*$  [72]. This property depends on the fact that the eigenvalues of the forward and adjoint operators are identical, which can be seen by first multiplying the balance statement by the adjoint eigenvector  $\phi_i^*$  and integrating,

$$\langle \phi_i^*, M\phi_i \rangle = \left\langle \phi_i^*, \frac{1}{\lambda_i} B\phi_i \right\rangle. \quad (3.35)$$

Note that a similar statement can be written from the adjoint balance statement. Solving for  $\lambda_i$  and applying the property of the adjoint, the equivalence of the forward and adjoint eigenvalues is clear [73]:

$$\begin{aligned} \lambda_i &= \frac{\langle \phi_i^*, B\phi_i \rangle}{\langle \phi_i^*, M\phi_i \rangle} \\ &= \frac{\langle B^* \phi_i^*, \phi_i \rangle}{\langle M^* \phi_i^*, \phi_i \rangle} \\ &= \eta_i \end{aligned} \quad (3.36)$$

Given that  $\lambda_i = \eta_i$ , the biorthogonality of the forward and adjoint flux can be seen by multiplying Eq. (3.34) by an arbitrary adjoint harmonic  $\phi_j^*$  and integrating :

$$\begin{aligned} \langle \phi_j^*, \hat{L}\phi_i \rangle &= \langle \hat{L}^* \phi_j^*, \phi_i \rangle \\ \langle \phi_j^*, \lambda_i \phi_i \rangle &= \langle \eta_j \phi_j^*, \phi_i \rangle \\ \lambda_i \langle \phi_j^*, \phi_i \rangle &= \eta_j \langle \phi_j^*, \phi_i \rangle \\ (\lambda_i - \eta_j) \langle \phi_j^*, \phi_i \rangle &= 0. \end{aligned} \quad (3.37)$$

This statement shows that if  $\lambda_i \neq \eta_j$ , then  $\langle \phi_j^*, \phi_i \rangle = 0$ , and that if  $\langle \phi_j^*, \phi_i \rangle \neq 0$ , then  $\lambda_i = \eta_j$ . Equation (3.37) therefore requires that  $\langle \phi_j^*, \phi_i \rangle = 0$  if  $i \neq j$ . A similar

biorthogonality statement may be derived for the expression

$$\langle \phi_j^*, L_i \phi_i \rangle = \langle \phi_j^*, M \phi_i \rangle - \left\langle \phi_j^*, \frac{1}{\lambda_i} B \phi_i \right\rangle, \quad (3.38)$$

by considering each term individually. By using the definition of the adjoint, the first term can be equated to  $\langle M^* \phi_j^*, \phi_i \rangle$ , which can in turn be equated to  $\frac{1}{\eta_j} \langle B^* \phi_j^*, \phi_i \rangle$  through the adjoint balance statement. Alternatively, the forward balance statement equates the first term to  $\left\langle \phi_j^*, \frac{1}{\lambda_i} B \phi_i \right\rangle$ , which, through the definition of adjoint, is  $\frac{1}{\lambda_i} \langle B^* \phi_j^*, \phi_i \rangle$ . This long chain of equalities therefore concludes with

$$\left( \frac{1}{\lambda_i} - \frac{1}{\eta_j} \right) \langle B^* \phi_j^*, \phi_i \rangle = 0, \quad (3.39)$$

which is clearly analogous to Eq (3.37) and results in the same conclusion, such that the second term in Eq (3.38) exhibits biorthogonality. Using the second term rather than the first in the above equalities, a similar statement may be made about the first term. Since Eq. (3.38) is a linear combination of these two terms, the full eigenvalue-dependent operator  $L_i$  also exhibits biorthogonality.

This biorthogonality of  $L$  motivates the following method for determining the values of the modal expansion coefficients in Eq. (3.33). Inserting a finitely-truncated modal expansion for  $\delta\phi$  into the first-order perturbation expression of Eq. (3.32) gives:

$$L \left( \sum_{i=0}^I a_i \phi_i \right) = -\delta L \phi, \quad (3.40)$$

where  $\delta L = \delta M - \frac{1}{\lambda_0} \delta B$  is the first-order approximation of the perturbed operator.

Multiplying by an arbitrary adjoint eigenvector  $\phi_j^*$  and integrating over all space and energy gives:

$$\left\langle \phi_j^*, L \left( \sum_{i=0}^I a_i \phi_i \right) \right\rangle = - \langle \phi_j^*, \delta L \phi \rangle. \quad (3.41)$$

Due to biorthogonality, all terms in the sum where  $i \neq j$  are zero, and it collapses to a single expansion coefficient. Since the  $\delta L$  term on the right hand side of Eq. (3.41) is arbitrary rather than determined by the balance equation, terms where  $j \neq 0$  are not necessarily zero, leaving an expression that may be readily solved for each  $a_j$  expansion coefficient:

$$\begin{aligned} \langle \phi_j^*, L(a_j \phi_j) \rangle &= - \langle \phi_j^*, \delta L \phi \rangle \\ a_j \langle \phi_j^*, L \phi_j \rangle &= - \langle \phi_j^*, \delta L \phi \rangle \\ a_j &= - \frac{\langle \phi_j^*, \delta L \phi \rangle}{\langle \phi_j^*, L \phi_j \rangle}. \end{aligned} \quad (3.42)$$

For  $j = 0$ ,  $L \phi_j = 0$  because the definition of  $L$  contains the fundamental eigenvalue, and therefore the denominator is zero and  $a_0$  is undefined. A special case of  $a_0 = 0$  is imposed. This is justified by considering that any  $\delta L$  that introduces a  $\delta \phi$  term in the shape of the unperturbed flux  $\phi = \phi_0$  will be treated by the flux normalization process on the perturbed result. As such, contributions from any arbitrary  $a_0$  term will be inconsequential, and  $a_0 = 0$  is chosen as a convenience.

The combination of Eq. (3.42) with Eq. (3.33) allows the determination of the perturbed flux  $\delta \phi$  as an algebraic expression of the pre-computed harmonics and the arbitrary perturbation  $\delta L$ , and thus provides a computationally-efficient method of

approximating the effects of any perturbation to the reference system. From the perturbed flux, a perturbed power distribution may be computed by normalizing the perturbed flux to the specified reactor power. With this information, a number of other core performance parameters may be evaluated for the perturbed state.

The perturbed adjoint flux is also of interest, as it allows the rapid computation of spatial reactivity coefficient distributions at the perturbed state through standard perturbation theory. Similar steps may be followed to determine the adjoint expansion coefficients that build  $\delta\phi^* = \sum_{i=0}^I a_i^* \phi_i^*$ . Following Eqs. (3.32) and (3.42), the expression for the adjoint expansion coefficients is:

$$\begin{aligned}
 L^* \delta\phi^* &= \delta L^* \phi^* \\
 \langle \phi_j, L^* (a_j^* \phi_j^*) \rangle &= -\langle \phi_j, \delta L^* \phi^* \rangle \\
 a_j^* &= -\frac{\langle \phi_j, \delta L^* \phi^* \rangle}{\langle \phi_j, L^* \phi_j^* \rangle}.
 \end{aligned} \tag{3.43}$$

### 3.1.3 Discussion of the Eigenvalue Expansion

Eigenmodes of the transport equation are not necessarily complete and cannot generally be used to represent the flux as a series expansion, but eigenmodes of various approximations of the transport equation (e.g. difference equations, P1, Diffusion) are known to be complete [72] and are therefore useful in Eq. (3.33), depending on their accuracy. It has traditionally been considered difficult to compute more than a few  $\lambda$ -eigenmodes of a large reactor simulation. The subtraction method can compute higher-order modes one at a time by subtracting previously-computed

lower-order harmonics during the power iteration. The direct method of computing the eigenvalues of the explicit operators  $L$  or  $Q$  has been considered impractical for large problems due to computational restrictions, though it has been attempted before. In some cases, great efforts were undergone to to efficiently determine the values of  $Q$  [74], where the direct method was shown to agree with the subtraction method. More recently, a subspace iteration has been used to determine the higher harmonics. Modak, et. al. discovered that a solution is guessed in the general shape of one of the eigenmodes, their Orthomin(1) solver would converge to it [75].

### 3.1.4 The Role of the Perturbed Eigenvalue

The change in the fundamental eigenvalue  $\lambda_0$  due to a perturbation of the reactor state is of critical importance for optimization studies. This change in  $\lambda_0$  associated with any perturbation does affect  $\delta L$ , and therefore does affect the perturbed flux. This can be understood by considering the numerator of the expression for the expansion coefficients in Eq. (3.42):

$$\begin{aligned} \langle \phi_j^*, \delta L \phi \rangle &= \left\langle \phi_j^*, \left[ \left( M' - \frac{1}{\lambda'_0} B' \right) - \left( M - \frac{1}{\lambda_0} B \right) \right] \phi \right\rangle \\ &= \left\langle \phi_j^*, \left[ \delta M - \frac{1}{\lambda_0 \lambda'_0} (\lambda_0 B' - \lambda'_0 B) \right] \phi \right\rangle \end{aligned} \quad (3.44)$$

In first-order MEPT, the approximation is made that  $\lambda'_0 = \lambda_0$  for building  $\delta L$ , resulting in

$$\delta L = \delta M - \frac{1}{\lambda_0} \delta B. \quad (3.45)$$

Once the perturbed flux  $\phi'$  has been approximated, the perturbed eigenvalue may be estimated explicitly by integrating the neutron balance statement over space:

$$\lambda'_0 = \frac{\langle B' \phi' \rangle}{\langle M' \phi' \rangle}. \quad (3.46)$$

However, since both the numerator and denominator contain first-order approximations, Eq. (3.46) turns out to be rather inaccurate. A preferred approach is to use standard perturbation theory to determine the perturbed eigenvalue, where

$$\Delta \rho = \frac{\langle \phi^*, \left[ \frac{1}{\lambda_0} \delta B - \delta M \right] \phi \rangle}{\langle \phi^*, B \phi \rangle}. \quad (3.47)$$

Equation (3.47) is substantially more accurate because it was designed to have second-order error by careful choice of  $\phi^*$ , and provides a means to compute the reactivity effect of any change in the reactor. This can be applied to reactivity coefficients of reactivity as well as understanding the criticality of a perturbed design for use in a criticality search, or other objectives.

Given the perturbed eigenvalue,  $\delta L$  and  $\delta \phi$  may be updated. In fact, the first-order truncation of Eq. (3.32) can be avoided altogether. If this is done, the denominator of Eq. (3.42) contains  $L'$  in place of  $L$  and the accuracy of MEPT can be improved.

### 3.1.5 MEPT In the Condensed Fission Source Problem

As discussed, the full eigenvalue problem is of order  $N * G$  but is only of rank  $N$ . Therefore, solving the condensed fission source problem allows a substantial gain in computational efficiency. The  $\delta L$  term of the full problem may be immediately estimated for any perturbation based on the imposed changes in number densities and/or microscopic cross sections. However, the analogous operator for the condensed problem from Eq. (3.23) is not a direct function of the design perturbations, as its creation involved the inversion of the  $A_g$  matrices during the inner iterations. We desire to use the easily-calculated  $\delta L$  from the full problem with harmonics computed through the condensed problem in order to determine the expansion coefficients.

Recall that in the full problem, the eigenvalue problem is comprised of  $\hat{L} = M^{-1}\chi F^T$  and  $\hat{L}\phi = \lambda\phi$ . In the condensed problem,  $Q = F^T M^{-1}\chi$  and  $Q\psi = \lambda\psi$ . The  $\hat{L}$  and  $Q$  operators are related by a similarity transformation in the one-group case,

$$\hat{L} = P^{-1}QP, \tag{3.48}$$

with  $P = M^{-1}\chi$ , and therefore have identical eigenvalues. In multigroup cases, the matrix  $P$  is rectangular and hence cannot have an inverse. Rather, we point out that the operators have identical rank (based on the number of fission source points in  $B$ ) and were constructed in §3.1.1 to solve the same problem. Since the condensed problem solution  $\psi$  implies the full solution  $\phi$ , we infer that any of the harmonics  $\phi_i$  may be similarly constructed, as all must satisfy the fundamental eigenvalue problem.

In the condensed problem, we solve Eq. (3.23) for the fission source harmonics  $\psi_j$  and then expand them to  $(N \times G)$  flux harmonics through an expression inspired by Eq. (3.25):

$$\phi_g^j = \frac{1}{\lambda_j} C_g \psi_j. \quad (3.49)$$

Of course, the  $C_g$  matrices are never formed in the computer code, and this expression is in practice solved by a set of computations identical to a single set of inner iterations by solving Eq. (3.30). One option is to store the full flux harmonics  $\phi_j$  and then recall them to generate the expansion coefficients for each particular  $\delta L$ , which in turn determine the perturbed flux. In this case, MEPT operates in  $N \times G$  vector space for the majority of operations, dropping to order- $N$  space only for the computationally-demanding matrix calculations of the fission source harmonics. Another option is to store only the fission source harmonics  $\psi_j$  and employ inner iterations to express each perturbation as the condensed operator  $\delta Q$ . The condensed operator can be determined through Eq. (3.28), obtaining

$$\delta Q \psi_j = Q' \psi_j - Q \psi_j = \sum_{g=1}^G F_g' C_g' \psi_j - \sum_{g=1}^G F_g C_g \psi_j. \quad (3.50)$$

In this equation, the second term is known from the reference calculation without additional work, and an inner iteration must be performed for each individual expansion coefficient. The perturbed fission source is computed in an expression analogous to Eq (3.42) with fission source harmonics in place of the expanded flux harmonics.

### 3.1.6 Computational Efficiency of MEPT

The motivation behind the development of MEPT is the substantial gain in computational efficiency of algebraic expressions over the direct solution of the diffusion equation. An informal discussion of the efficiency follows, as the iterative nature of the algorithms at hand complicates a formal discussion. Once a reference solution to a problem with  $NG = N \times G$  mesh points is computed with  $I$  harmonics, the work required to determine the flux of a design perturbation with MEPT includes:

- $O(NG)$  subtractions to compute  $\delta L$  in Eq. (3.16)
- $I \times (NG \times (G + 6))/2$  additions/multiplications to evaluate the  $I$  expansion coefficients given  $\delta L$  in Eq. (3.42)
- $I \times O(NG)$  additions/multiplications to combine the harmonics into the perturbed flux in Eq. (3.33)
- $O(NG)$  additions/multiplications to perform power normalization

For comparison, a full power iteration undergoing  $K$  inner iterations requires

- $K \times \left( \frac{G(G+1)}{2}N + G \times N \right)$  additions/multiplications to solve Eq. (3.30)
- $O(N)$  additions to update the eigenvalue during the outer iterations in Eq. (3.27)
- $O(NG)$  additions/multiplications to perform power normalization

Asymptotically, MEPT requires  $I \times O(G^2N)$  operations and direct methods require  $K \times O(G^2N)$ .

Done naively, the  $K$  inversions of the each  $A_g$  matrix would require  $O(N^3)$  operations. In practice, the iterative methods that are used in the inner iterations require fewer than  $O(N^2)$  operations per iteration, and are often closer to  $O(N)$  because of the sparsity of the matrices. More advanced methods such as conjugate gradients (CG) and generalized minimal residual (GMRES) may improve upon successive over-relaxation (SOR), but are still  $O(N^2)$  [76]. The computational advantage of MEPT over direct methods on a single processor depends on the balance between the number of harmonics required and the number of total inner iterations. It is not uncommon to require between 20 and 200 outer iterations for a case to converge in industrial power iteration codes, depending on its dominance ratio. Without acceleration based on source extrapolation, the power method for practical cases may require up to 1000 iterations to converge to  $\epsilon = 1 \times 10^{-8}$ . The number of inner iterations per outer iteration has been discussed theoretically [77, 78, 79], and in practice is often between 2 and 20, although it can sometimes be much higher [80]. However, when a reference state is used as the initial guess, the number of outer iterations will likely remain on the order of 20.

The major computational advantage of MEPT is evident when considering parallel processing. Direct methods necessarily iterate, but the expansion coefficients from MEPT are completely independent and may be computed concurrently on a parallel computer. Where the advantage of MEPT on a single CPU is only substantial when many inner iterations are required, the speedup factor offered by MEPT on a parallel computer is roughly equal to the number of processors available.

### 3.1.7 Parameters Available Through MEPT

Given perturbed multigroup flux and adjoint flux shapes from a MEPT calculation, normalization yields a perturbed power distribution. Once perturbed flux and power fields from MEPT are loaded into the abstract reactor model of ARMI as described in §2.2.1, all other modules may run transparently. To ARMI, MEPT is just another neutronics kernel. Accordingly, the thermal hydraulics, fuel performance, cladding wastage, duct distortion, economics, and other modules may be executed to update the perturbed state without further development.

For perturbed safety analysis, spatial reactivity worth distributions may be updated using first-order perturbation through Eq. (3.47). These reactivity worth distributions are essential input to transient analysis codes such as SAS4A/SASSYS [67] which may be executed to estimate the perturbed transient performance in a variety of accident scenarios. Complementary to the reactivity worth distributions, the effective delayed neutron fraction may be approximated, since all perturbed terms in the adjoint-weighting expression,

$$\beta = \frac{\langle \phi^*, \chi_d \nu_d \Sigma_f \phi \rangle}{\langle \phi^*, \chi \nu \Sigma_f \phi \rangle}, \quad (3.51)$$

(where  $\chi_d$  and  $\nu_d$  are the delayed neutron emission spectrum and yield) are available. For these reactivity coefficients, the perturbed flux is obtained with expansion coefficients from Eq. (3.42) and the perturbed adjoint flux is obtained through Eq. (3.43).

The transmutation rates of the nuclides in the core are perturbed by the flux

change of any design modification. The consequences of this fact are discussed at length in Chapter IV.

### 3.1.8 Arnoldi's Method for Calculating the Modal Harmonics

Each MEPT problem requires two eigenvalue problems (forward and adjoint) to be solved while retaining many harmonics. More localized perturbations will require larger numbers of harmonics than more global perturbations. Typical nuclear reactor simulations are only interested in the dominant eigenvalue,  $k_{eff}$ . As such, the power method is typically used to solve Eq. (3.23). Since MEPT requires multiple eigenvalues and harmonics, more sophisticated eigenvalue solvers are required. Classic eigenvalue solvers produce an upper Hessenberg matrix directly, and then iteratively transform it to triangular form, thus undergoing a Schur factorization, where the eigenvalues are found on the diagonal of the triangular matrix. These solvers quickly become impractical for the large sparse systems that arise in the discretization of the equations governing nuclear reactors. Traditional triangularization methods such as Householder reflections would destroy the sparsity of these matrices, requiring substantially more memory than is required to hold the problem. We naturally come to the class of iterative solvers, called Krylov subspace methods, that require only the action of a matrix on a vector to efficiently approximate the eigenvalues and eigenfunctions. Sparse matrices with  $\nu$  non-zero entries per row can produce the product  $Ax$  with  $O(\nu m)$  operations compared to  $O(m^2)$  required for dense matrices [81].

For large systems, iterative eigenvalue solvers have been put to highly-successful use. Arnoldi's method is an iterative Krylov method to determine the eigenvalues

and eigenvectors of a matrix  $A$ . We review the algorithm based on Trefethen and Bau [81]. It is an algorithm to reduce a nonhermitian matrix to Hessenberg form by orthogonal similarity transformations proceeding column by column from a prescribed first column  $q_1$ .

A Hessenberg reduction is defined as  $A = QHQ^*$ , where  $H$  is upper triangular above the subdiagonal and  $Q$  is unitary. For iterative methods, we consider  $m$  to be infinite, so we consider only the first  $n$  columns of  $Q$ :

$$Q_n = \begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix}. \quad (3.52)$$

$\hat{H}_n$  is the  $(n+1) \times n$  upper-left section of  $H$ , which is also a Hessenberg matrix.

Then,

$$AQ_n = Q_{n+1}\hat{H}_n, \quad (3.53)$$

$$\begin{bmatrix} A \end{bmatrix} \begin{bmatrix} q_1 & \dots & q_n \end{bmatrix} = \begin{bmatrix} q_1 & \dots & q_{n+1} \end{bmatrix} \begin{bmatrix} h_{11} & \dots & h_{1n} \\ h_{21} & \ddots & \\ & & h_{n+1,n} \end{bmatrix}. \quad (3.54)$$

The  $n$ -th column of Eq. (3.54) may be written as:

$$Aq_n = h_{1n}q_1 + \dots + h_{nn}q_n + h_{n+1,n}q_{n+1}, \quad (3.55)$$

where it is clear that  $q_{n+1}$  satisfies an  $(n+1)$  recurrence relation with the previous Krylov vectors. The Arnoldi iteration is analogous to the modified Gram-Schmidt

---

**Algorithm III.1** The Arnoldi Iteration

---

$b$ =arbitrary,  $q_1 = \frac{b}{\|b\|}$   
for  $n=1,2,3,\dots$

$$v = Aq_n$$

for  $j=1$  to  $n$

$$h_{jn} = q_j^* v$$

$$v = v - h_{jn} q_j$$

$$h_{n+1,n} = \|v\|$$

$$q_{n+1} = \frac{v}{h_{n+1,n}}$$

---

iteration that solves Eq. (3.55), shown explicitly in Algorithm III.1.

At each iteration of the Arnoldi iteration, the eigenvalues of the reduced the Hessenberg matrix  $\hat{H}_n$  may be computed using standard direct methods. Because  $H_n$  is similar to  $A$ , the eigenvalues of  $\hat{H}_n$  (known as the Ritz values) are approximations of the eigenvalues of  $A$ . If the approximations are good, the problem of computing the first  $n$  eigenvalues of an  $m \times m$  matrix  $A$  is reduced to a problem of computing the eigenvalues of the  $n \times n$  matrix  $\hat{H}_n$ . As  $m$  is very large, the gains in computational efficiency are potentially immense. The Arnoldi method will find the eigenvalues on the edges of the eigenvalue spectrum, which are, in this application, the ones of most interest.

Several numerical issues with the standard Arnoldi algorithm were solved in the implicitly-restarted Arnoldi method (IRAM) [82], as implemented in ARPACK [83]. Each additional column added to the Krylov subspace required more storage, leading to memory issues for large problems. However, this work uses a more recent implementation from Trilinos [84], called the Krylov-Schur algorithm, which is mathemati-

cally identical to IRAM, but offers improved performance in deflating converged Ritz vectors from the decomposition and avoids a potential instability of the implicit QR algorithm that can produce unwanted Ritz vectors [85].

We are driven to Arnoldi’s method as a means to compute multiple harmonics for the flux expansion in Eq. (3.33), but others have moved towards Krylov-based solvers in nuclear reactor simulation for separate reasons. Verdu [86] found IRAM superior to several other methods of computing the first few eigenvalues of light-water reactors. Warsa [54] found IRAM to be substantially more efficient than the power iteration in high dominance-ratio neutron transport criticality problems. Recently, Cao [80] implemented ARPACK with iterative inner iterations in 2-D to compute the  $\alpha$ -eigenmodes of accelerator-driven subcritical assemblies.

### 3.1.9 Relation to Other Methods

The most related work in the literature to MEPT is the high-order FRENCH method [28], based on the so-called Standard Method (SM) [24]. In this method, the perturbation  $\delta L$  is expressed as

$$\delta L = \delta M + \lambda'_0 \delta B + \delta \lambda_0 B, \tag{3.56}$$

where  $\lambda_0$  is the reciprocal of the fundamental eigenvalue (note that in this section, this symbol has been changed from previous sections for simplicity in the derivation) and the perturbed flux is expressed as a series of flux terms of successively higher order:

$$\phi' = \sum_{i=0}^{\infty} f^{(i)}. \quad (3.57)$$

A recursion may be obtained from the expression  $L'\phi' = 0$  where one first determines the lowest-order term  $f^{(0)}$  and then computes each higher-order term, one by one. Considering only the zeroth-order terms, we obtain the unperturbed balance statement  $Lf^{(0)} = 0$ , where it is clear that  $f^{(0)} = \phi$ . Given the value of  $f^{(0)}$ , the expression involving only zeroth- and first-order terms can be solved for  $f^{(1)}$ :

$$Lf^{(0)} + Lf^{(1)} + \delta Lf^{(0)} = 0. \quad (3.58)$$

In each higher-order expression, the lower-order terms, such as  $Lf^{(0)}$  in Eq. (3.58), cancel out due to the previous order. To be clear, the second-order expression contains all terms up to second order from the expression  $(L + \delta L)(f^{(0)} + f^{(1)} + f^{(2)}) = 0$ , or

$$Lf^{(0)} + Lf^{(1)} + Lf^{(2)} + \delta Lf^{(0)} + \delta Lf^{(1)} = 0, \quad (3.59)$$

where we see that applying Eq (3.58) results in

$$Lf^{(2)} + \delta Lf^{(1)} = 0. \quad (3.60)$$

Continuing, each higher order follows the recurrence relation

$$Lf^{(i)} = -\delta Lf^{(i-1)}. \quad (3.61)$$

The perturbed eigenvalue is expanded similarly, as

$$\lambda'_0 = \sum_{i=1}^{\infty} \lambda_0^{(i)}, \quad (3.62)$$

where the superscript  $i$  again represents the term order. The recurrent system of Eq. (3.61) can be expressed considering Eq. (3.56) with the eigenvalue expansion to determine  $f^{(i)}$  :

$$(M + \lambda_0 B) f^{(0)} = 0 \quad (3.63)$$

$$(M + \lambda_0 B) f^{(i)} = - \left\{ \delta M f^{(i-1)} + \sum_{k=0}^{i-1} \left[ \lambda_0^{(i-k-1)} \delta B + \lambda_0^{(i-k)} B \right] f^{(k)} \right\} \quad (3.64)$$

Rather than solving the recurrent system for each individual perturbation, the SM considers each  $f^{(i)}$  as an expansion in terms of the unperturbed eigenfunctions

$$f^{(i)} = \sum_{j=0}^{\infty} a_j^{(i)} \phi_j. \quad (3.65)$$

Substitution of this expansion into the recurrent system followed by multiplication by an arbitrary adjoint harmonic  $\phi_n^*$ , and then integration over space and the application of biorthogonality exposes expressions that can be solved for each coefficient  $a_j^{(i)}$ .

Both MEPT and the SM expand the perturbed flux on the basis of reference eigenmodes and use adjoint biorthogonality to determine expansion coefficients that lead to an explicit representation of the perturbed flux. The SM requires two levels of expansion, one in term order and the other in the number of harmonics, whereas MEPT

only requires the expansion in harmonics. When expansion coefficients from the SM are expressed for first-order for comparison with Eq. (3.42) in MEPT, Palmiotti found

$$a_n^{(1)} = \frac{1}{\lambda_1 - \lambda_0} \left[ \sum_{j=0}^{\infty} \frac{\langle \phi_n^*, (\delta M + \lambda_0 \delta B) \phi_j \rangle}{\langle \phi_n^*, F \phi_n \rangle} \right] \quad (3.66)$$

The expansion coefficients for the SM require the evaluation of a matrix of inner products of the flux harmonics, whereas the MEPT formulation requires only a single pair of inner products for each harmonic. These inner products are not nearly as computationally demanding as a direct solve of a system, but with hundreds of harmonics, the added burden of the SM will make the method noticeably slower. Additionally, the coupled nature of the harmonics precludes parallel evaluation, which is a computational disadvantage compared to MEPT. Also, as mentioned in §1.3.2, the SM was found to perform very poorly for multi-group problems and was later extended (becoming the FRENCH method) through the use of group-wise expansions and heuristic orthogonality statements. The results from the FRENCH method shown in [28] prove that the method does in fact work well for large perturbations in 2-D, 3-group problems.

### 3.2 Implementation of MEPT within the ARMI Framework

MEPT has been implemented in a new multi-group finite-difference diffusion code based on the DIF3D formulation called DIFNT. The code has been written as an ARMI module, enabling close coupling with all other ARMI modules and their

associated capabilities.

The full problem in Eq. (3.15) was originally studied for small 1-D cases with direct solvers to prove the feasibility of MEPT and for comparison with the condensed fission source problem of Eq. (3.23). For larger cases, more powerful libraries are called upon, relying exclusively on subspace-based iterative methods.

### 3.2.1 Development of the New Diffusion Code DIFNT

The DIFNT code is developed to solve realistic reactor problems. The regular triangular mesh can be specified at arbitrary resolution, with 6, 24, 96, or more mesh points per ARMI block, as allowed by memory. The axial mesh is uniform but may be irregular. DIFNT is tightly integrated into ARMI; it shares geometry and composition loading files and passes output directly to the abstract reactor model without using intermediate transfer files.

A prototype version of DIFNT was written for Cartesian geometry in Python [87]. The code performance was acceptable for 1- and 2-D test problems, but for more realistic 3-D problems with more than ten energy groups, the prototype became impractically slow. To enable ARMI integration while solving problems efficiently, DIFNT is written in hybrid C++/Python. Once the block-level macroscopic cross sections are prepared on the Python side using the reactor specifications of the ARMI model, the C++ extension activates and reads the macroscopic cross sections directly through the Python-C API (no use of wrappers such as SWIG are employed). These data are assembled into data structures compatible with the Trilinos C++ matrix library [84, 88]. Trilinos functions are relied upon for the subsequent matrix manip-

ulations involved in formulating and solving Eq. (3.23). Specifically, the AztecOO package [89] using either the CG or GMRES solver is implemented to perform the inner iterations of Eq. (3.30) at each step of the outer iterations. The outer iterations of Eq. (3.26) are solved with the Block Krylov Schur method of the Amesos package [90]. An specialized `Epetra_Operator` object with the inner iterations embedded in it was implemented to compute the implicit action of  $Q$  on an arbitrary vector  $\psi$ . The code retains as many Ritz pairs as requested by the user for later use in computing expansion coefficients for perturbations. Trilinos was chosen as a matrix library due to its wide selection of modules and its previous success being parallelized for neutronics simulations [91]. The potential to parallelize the DIFNT solvers preserves the possibility of handling extremely large problems efficiently, although this is not attempted in this work.

DIFNT has been built as a Python extension on both MS Windows © and Linux platforms, using gcc 4.6.1 and Microsoft Visual C++ 2008. The build process combining Trilinos, Windows, and Python was particularly challenging. Detailed notes on the process are available by request.

### 3.2.2 Validation of DIFNT Against DIF3D

To validate DIFNT, ARMI cases were executed with both kernels using the same microscopic cross section set. The test case is a 3-D bare reactor with 12% enriched U-10Zr fuel at 75% smear density, modeled in 1/3 symmetry. It has 178 fuel assemblies, each split evenly into 3 axial blocks. The results for both kernels are shown in Table 3.1. The codes produce identical k-eigenvalues, with very small differences oth-

Table 3.1: DIFNT results compared with DIF3D reference

	DIFNT	DIF3D
$k_{eff}$	1.027035	1.027035
Peaking Factor $F_q$	3.643	3.643
Peak Power Density (W / cm <sup>3</sup> )	688.036	688.033
Fissile Production Rate (# / s)	$3.0665 \times 10^{19}$	$3.0661 \times 10^{19}$
Neutron Absorption Rate (# / s)	$2.5460 \times 10^{19}$	$2.5456 \times 10^{19}$
(n,2n) Production Rate (# / s)	$1.5849 \times 10^{17}$	$7.9233 \times 10^{16}$

erwise, the most notable being in the (n,2n) production rate. However, the DIFNT result is exactly two times the DIF3D value, suggesting a printout error in DIF3D. Numerical experiments verify that the codes are implemented identically in regard to the (n,2n) reaction, showing that substantial changes in  $k_{eff}$  arise when the (n,2n) cross sections are halved to match the DIF3D (n,2n) rate. The 33-group flux spectra shown in Figure 3.3 are graphically identical. The relative values of each energy group differ by  $-0.014 \pm 0.00002\%$ , suggesting a small systematic difference in the power normalization step.

The validation results suggest that the Trilinos-based Krylov outer iteration is in strong agreement with the traditional power iteration as implemented in DIF3D. Besides potentially being parallelized, DIFNT has the additional capability to return as many  $\lambda$ -eigenmodes as are requested in a numerically well-behaved manner.

### 3.2.3 Using DIFNT to Calculate Modal Perturbation Coefficients

The determination of perturbed flux distributions through Eq. (3.42) requires a number of real and adjoint  $\lambda$ -eigenmodes  $\phi_j$  and  $\phi_j^*$  to be known. DIFNT calculates and stores these harmonics while solving for the fundamental eigenmode  $\phi_0$  and its

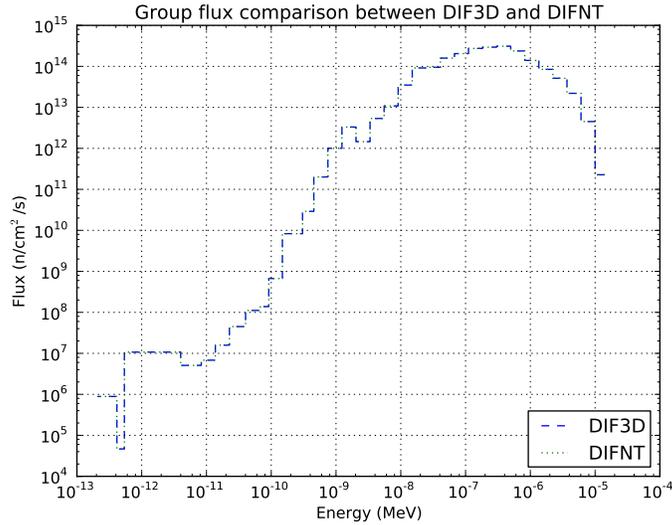


Figure 3.3: The 33-group flux spectrum computed by DIFNT compared to that of DIF3D.

associated eigenvalue  $\lambda_0$ . The first twenty eigenmodes of a sample problem are shown graphically in Figure 3.4. The energy dependence of each harmonic is in magnitude only, as each group was expanded from the same group-independent fission source harmonic. The first harmonic is the fundamental mode in the shape of the flux distribution. The second is the first harmonic in the axial direction, with negative values towards the top and positive values at the bottom. The radial shape throughout is similar to the fundamental mode. The third harmonic is radially bimodal, with positive values on the left and negative values on the right with an axial dependence similar to the fundamental mode.

The mesh size of the finite-difference approximation determines the maximum number of eigenvalues that can be computed; a case with  $N$  non-zero mesh points will have a maximum of  $N$  independent eigenmodes. The 90 largest real eigenvalues

of the sample problem are plotted sequentially in Figure 3.5. The problem has 1014 mesh points, so this plot represents about 10% of the total eigenvalues.

In perfectly symmetric problems (typically fresh cores), particularly in Cartesian geometry, degenerate eigenmodes may arise. Easily detected by their identical eigenvalues, these eigenmodes are not biorthogonal as required in MEPT. The method can still operate if the repeated values are dropped, but an undesirable randomness enters the results depending on which particular linear combination of the degenerate eigenmodes is returned by the solver. One combination may capture a particular perturbation better than another. Possible approaches to detect the optimal combination for a given perturbation have been explored, but the issue is rare in practical calculations, which almost always have some degree of asymmetry.

### **3.3 Calculating Perturbed Power and Flux Distributions with MEPT**

In this section, the DIFNT code is exercised to approximate the results of various perturbations at a particular point in time. The validation case from §3.2.2 is used as the references case. Localized perturbations are focused upon as they are the most challenging to determine, having excited higher flux harmonics.

#### **3.3.1 Performance of MEPT with Perturbations in an Assembly Cluster**

A cluster of seven neighboring assemblies in an asymmetric location of the core is perturbed by uniformly increasing the uranium enrichment by 20%, from 12% U-235 to 14.4% U-235. The enrichment distribution of the perturbed core is shown in

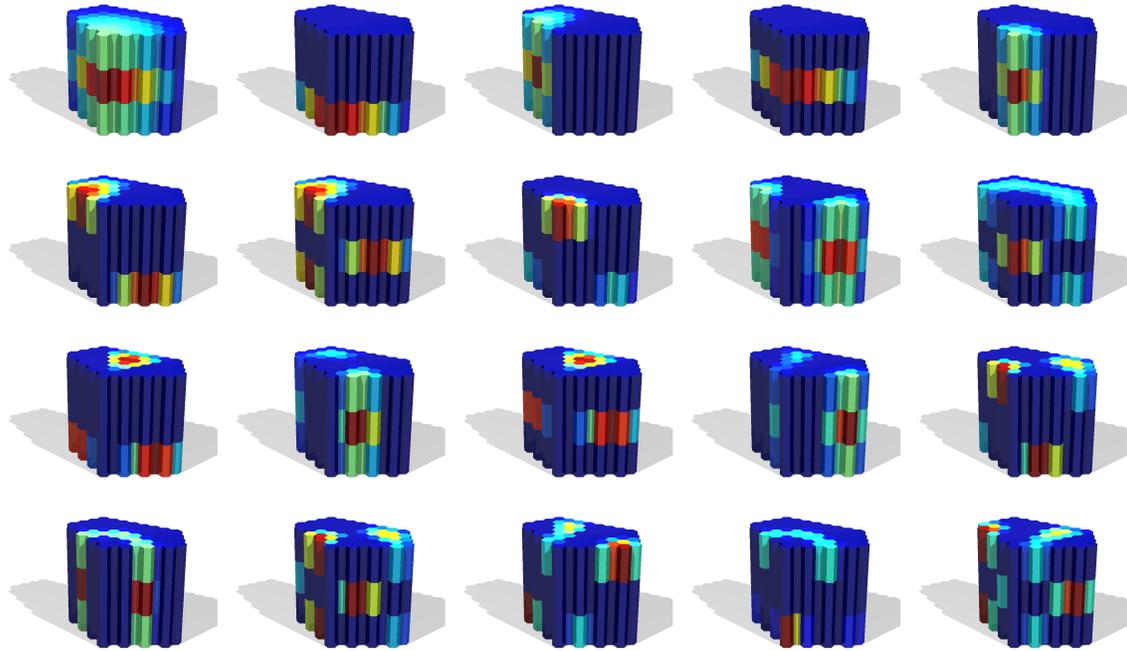


Figure 3.4: The first 20 real flux harmonics of a bare reactor sample case as calculated by DIFNT. The harmonics are summed over all 33 energy groups.

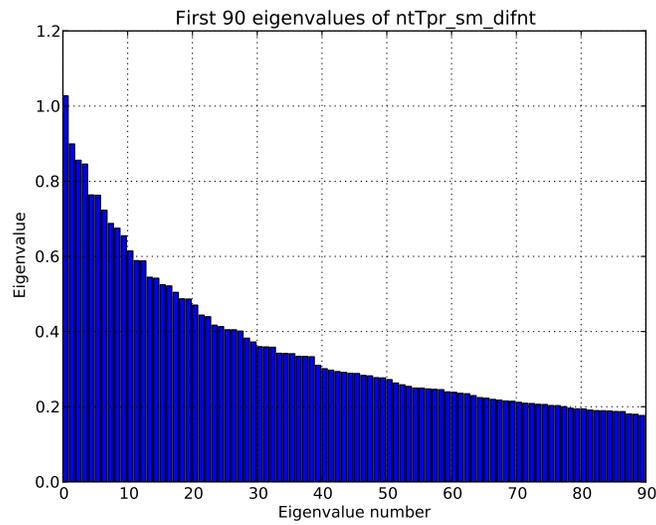


Figure 3.5: The 90 largest real eigenvalues of a bare sample problem as calculated by DIFNT.

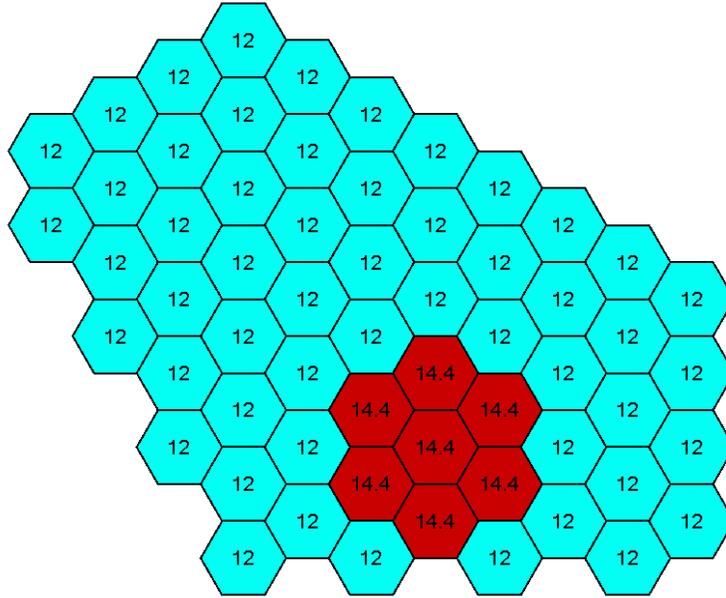


Figure 3.6: The uranium mass enrichment distribution (in %) of the perturbed cluster of assemblies.

Figure 3.6.

After computing the first 400 harmonics, the DIFNT code was executed in perturbation mode to determine the expansion coefficients corresponding to each design perturbation through Eq. (3.42). The coefficients for the largest perturbation are shown in Figure 3.9. The expansion coefficients from the less perturbed cases are proportionally smaller in magnitude. The success of MEPT in approximating localized perturbations is clearly demonstrated by comparing the fractional flux deviation between the reference case and the directly-calculated perturbed case in Figure 3.7 with that between the direct case and the perturbed case computed by MEPT in Figure 3.8.

The spectra of the coefficients between the various perturbation magnitudes are



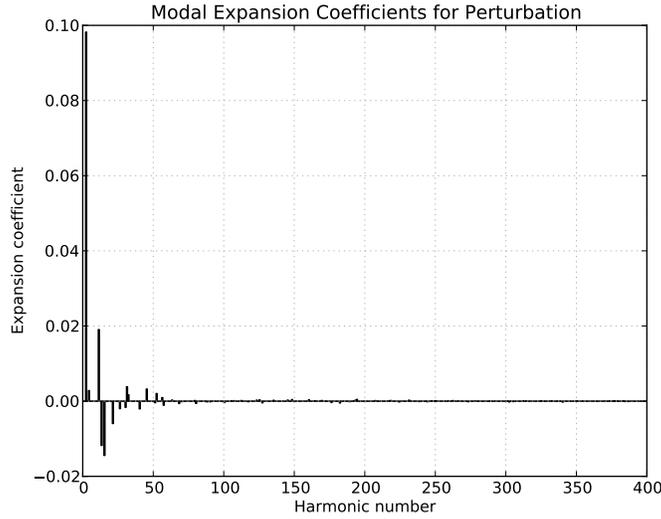


Figure 3.9: The first 400 expansion coefficients for a 20% increase in uranium enrichment in a cluster of assemblies.

very similar, but not identical. Perturbed solutions were calculated considering various numbers of harmonics and compared to a reference solution computed by running DIFNT directly at the perturbed state. The maximum deviation from the reference is shown for four different perturbations and for various numbers of harmonics in Figure 3.10. The convergence with number of harmonics is not monotonic, as one harmonic may contribute before another of similar magnitude that will cancel the larger deviations is added. To determine the number of harmonics required to capture a particular class of perturbation, one may start with many and reduce the number based on a convergence study if desired. Since the harmonics calculation only occurs once in a larger set of perturbation cases, it is not a substantial computational disadvantage to compute many harmonics. Peak and average flux errors are shown for the smallest magnitude perturbation in Figure 3.11. While the core-average error

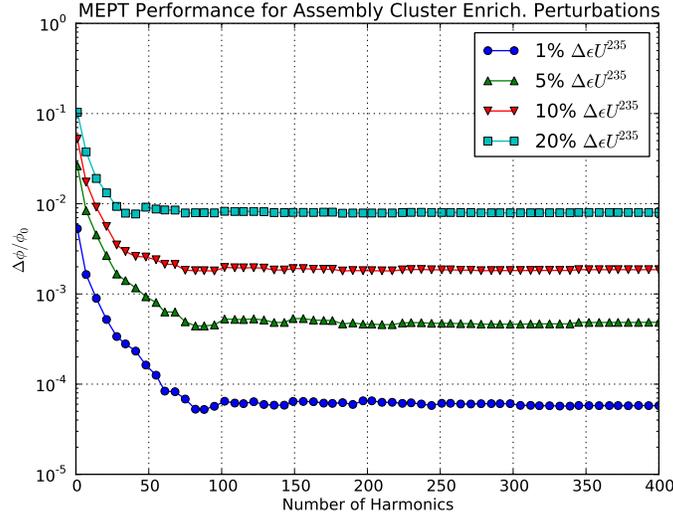


Figure 3.10: The core-average fractional point-flux deviation between perturbed solutions computed by DIFNT and a reference solution considering increasing numbers of harmonics for an offset cluster of assemblies.

converged after 100 harmonics were considered, the peak error continues to improve until 300 harmonics are considered in this case.

Larger perturbations result in proportionally worse uncorrected and corrected deviations from the reference. The 20% enrichment perturbation experienced a peak point-wise error reduction from 20.6% to 2.1%, while the 1% enrichment perturbation experienced a reduction from 1.1% to 0.0091%. The fact that smaller perturbations lead to less error is intuitive.

The fundamental eigenvalue is altered by 2600 pcm from the reference state in the 20% perturbation. When computed by the explicit estimator from Eq. (3.46), the resulting perturbed eigenvalue deviates from the directly calculated result by 320 pcm with 400 harmonics considered. The standard perturbation expression in Eq. (3.47)

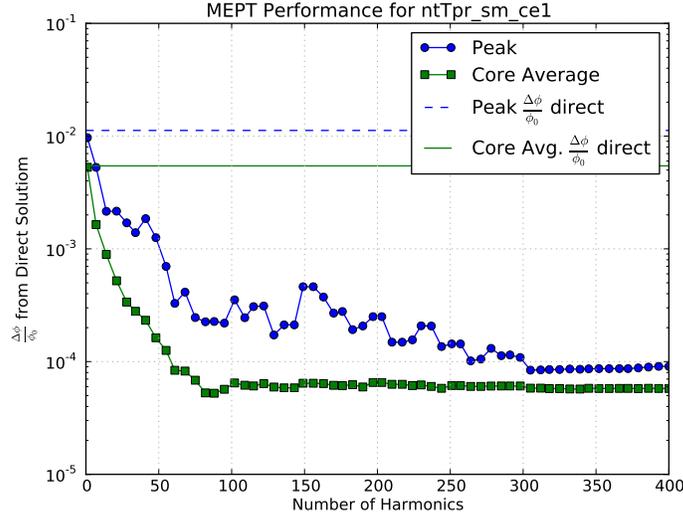


Figure 3.11: Core-average and point fractional flux deviations of 1% enrichment perturbation on an offset cluster of assemblies.

differs from the directly-calculated eigenvalue by 116 pcm. The effect of updating  $\delta L$  with the correct perturbed eigenvalue for both the first- and second-order MEPT formulations may be seen in Figure 3.12. Using the unperturbed eigenvalue, the second-order approximation performs substantially worse than the first-order because the second-order form of Eq. (3.42) contains approximate  $\delta L$  terms in both the numerator and denominator, whereas otherwise it is only found in the numerator. However, if  $\lambda'_0$  is known, the second-order approximation outperforms the first, as expected.

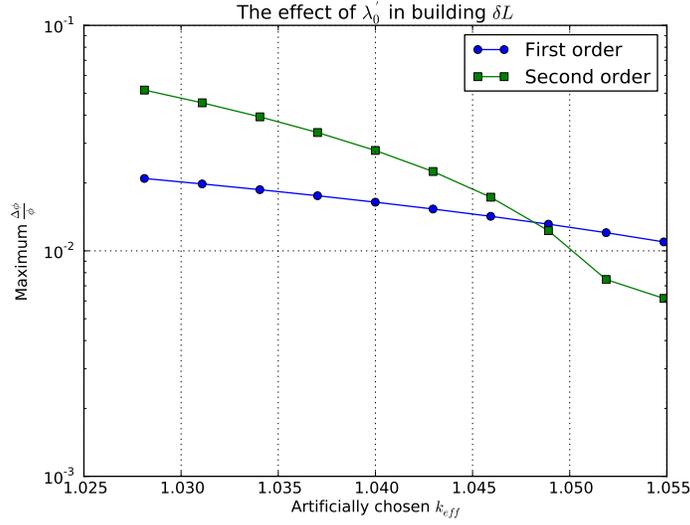


Figure 3.12: Maximum point flux deviations from reference of the 1st- and 2nd-order MEPT formulations with  $\lambda'_0$  at various points between  $\lambda_0$  and the directly-calculated perturbed eigenvalue.

### 3.3.2 Performance of MEPT with Modeling Perturbations to a Single Assembly

A smaller but more localized perturbation is investigated by adjusting the enrichment of a single assembly in the fifth radial ring. The expansion coefficients resulting from the 20%-increase in enrichment case are shown in Figure 3.13, where comparison with those from Figure 3.9 demonstrates the expected result that more localized perturbations result in larger relative importance of the higher harmonics. The convergence of the single-assembly perturbed cases as additional harmonics contribute is shown in Figure 3.14.

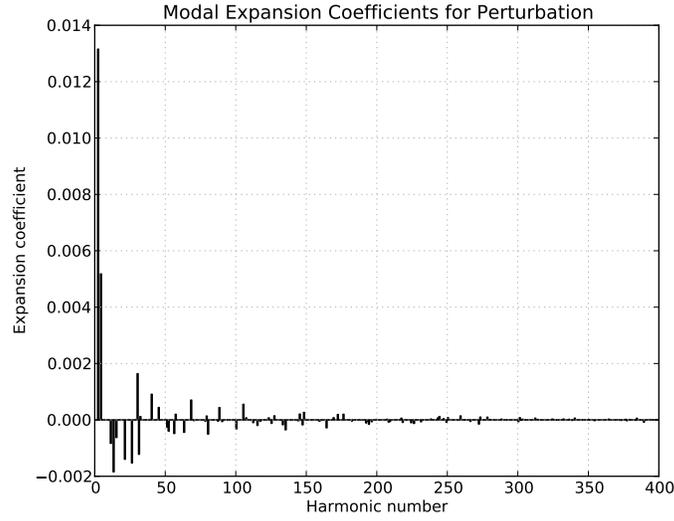


Figure 3.13: The first 400 expansion coefficients for a 20% increase in uranium enrichment in a single assembly in ring position 5.

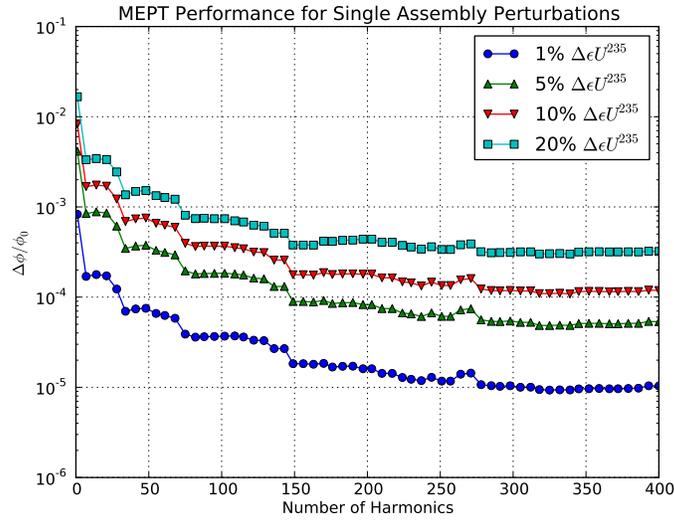


Figure 3.14: The core-average fractional point-flux deviation between single-assembly perturbed solutions computed by DIFNT and a reference solution computed by directly solving the diffusion equation for the perturbed state considering increasing numbers of harmonics.

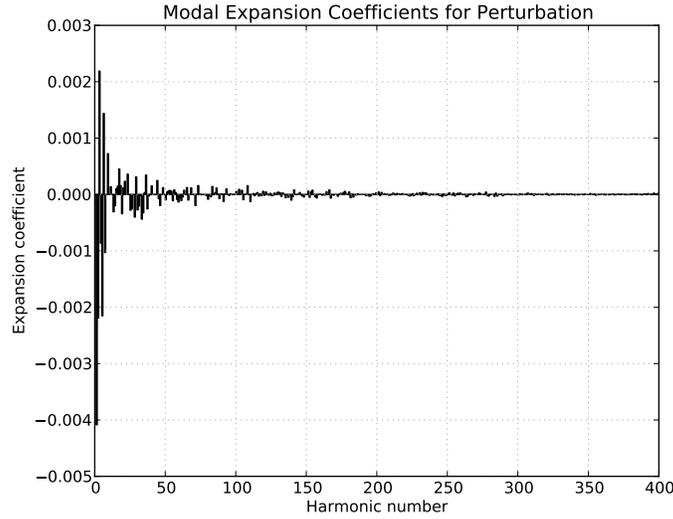


Figure 3.15: The first 400 expansion coefficients for a 20% increase in uranium enrichment in the bottom block of a single assembly in ring position 5.

### 3.3.3 Performance of MEPT with Modeling Perturbations to a Single Block of a Single Assembly

A substantially localized perturbation of the bottom third of an assembly in the 5th ring is studied for comparison with the previous two larger perturbations, with expansion coefficients resulting from the 20%-increase case shown in Figure 3.15. Again, the more localized and asymmetric perturbation leads to additional relative excitation of higher harmonics, as well as increased excitation density of harmonics. The effects of adding more harmonics for the single-block perturbed cases are shown in Figure 3.16. Since this is a smaller overall perturbation to the system, the largest magnitude of any expansion coefficient is smallest in this case, as is the uncorrected change in flux.

The three localized perturbations shown in this section demonstrate that MEPT

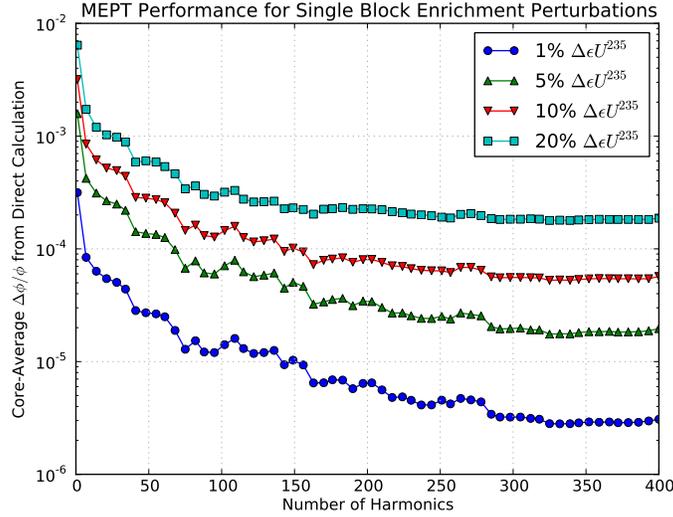


Figure 3.16: The maximum fractional point-flux deviation between single-block perturbed solutions computed by DIFNT and a reference solution computed by directly solving the diffusion equation for the perturbed state considering increasing numbers of harmonics.

is indeed capable of approximating the effects of asymmetric, highly localized perturbations of substantial magnitude. The explicit nature of MEPT that produces core-wide, multigroup flux distributions allows the consideration of many aspects of core design through data management systems such as the ARMI, as opposed to traditional variational methods which determine the effects of perturbations on one functional of interest at a time. This capability enables MEPT to be applied to equilibrium cycles as required for the design optimization of high-burnup fast reactors.

## CHAPTER IV

# Equilibrium Cycle Perturbation Theory

In this chapter, an Equilibrium Cycle Perturbation Theory is developed to allow the rapid evaluation of the effects of design perturbations on the equilibrium state of a reactor core. The method adds a treatment of nuclide depletion that works in concert with Modal Expansion Perturbation Theory as developed in Chapter III to produce perturbed nuclide and neutron fields. These methods are implemented within the Enhanced Equilibrium Cycle discussed in Chapter II, maintaining the advances made there. Supporting the goal of developing an efficient and extensible tool, the equilibrium treatment developed in this chapter optionally contains a first-order acceleration of nuclide depletion.

### 4.1 A First-Order Depletion Approximation

The perturbed flux capabilities of MEPT can be augmented by a nuclide density perturbation theory to propagate design changes through the cycle. A microscopic depletion scheme that explicitly tracks the actinide densities with time is preferable

to the macroscopic schemes used in many LWR codes due to its ability to track history effects, which are very important for closed-cycle and high-burnup reactors. The time rate of change of the number density  $N_i$  of nuclide  $i$  in a microscopic depletion scheme may be written as the Bateman equation, repeated from Eq. (2.1):

$$\frac{\partial N_i}{\partial t} = -N_i \left( \sum_{g=1}^G \sigma_{a,g}^i \phi_g + \lambda_i \right) + \sum_{j=1}^I N_j \left( \sum_{g=1}^G (\gamma_g^{j \rightarrow i} \phi_g) + \lambda_{j \rightarrow i} \right), \quad (4.1)$$

where

$\sigma_{ag}^i$  is the  $g$ -th group absorption cross section for nuclide  $i$ ,

$\lambda_i$  is the decay constant for nuclide  $i$ ,

$\gamma_g^{j \rightarrow i}$  is the transmutation yield from nuclide  $j$  to nuclide  $i$ , and

$\lambda_{j \rightarrow i}$  is the decay constant from nuclide  $j$  to nuclide  $i$ .

This equation may be written concisely in matrix notation as

$$\frac{\partial N}{\partial t} = AN, \quad (4.2)$$

where  $A$  is the transmutation matrix and  $N$  is a vector of nuclide densities. The time-dependent solution to Eq. (4.2) is a matrix exponential,

$$N(t) = e^{A(\phi)t} N_0 = \left[ I + At + \frac{(At)^2}{2!} + \dots \right] N_0. \quad (4.3)$$

In reactor codes, a nuclide vector and transmutation matrix are stored on each spatial region.

### 4.1.1 Derivation of the First-Order Depletion Approximation

Since the transmutation matrix is a function of the flux through Eq. (4.1) and the flux is a function of the nuclide densities through Eq. (3.15), the nuclide and neutron fields are inherently coupled. Thus, any design perturbation that results in a  $\delta\phi$  is accompanied by a  $\delta A$  and the corresponding change in the nuclide depletion rates. Design perturbations are expressed with a possible change in nuclide densities  $\delta N_0$  and the associated perturbation in macroscopic cross sections and flux. For a design change  $\delta N_0$ , we seek a perturbation-based expression for its effect on the quasi-static reactor model. The changes that arise from such a perturbation may be expressed through Eq. (4.3) as:

$$\begin{aligned}
 N'_0 &= N_0 + \delta N_0 \\
 \phi'_0 &= \phi_0 + \delta\phi_0 \\
 A_0(\phi'_0) &= A_0 + \delta A_0 \\
 e^{A'_0 t} &= I + (A_0 + \delta A_0)t + \frac{(A_0 + \delta A_0)^2 t^2}{2!} + \dots \\
 &= e^{A_0 t} + \delta A_0 t + \frac{(2A_0 \delta A + \delta A_0^2) t^2}{2!} + \dots \\
 N'(t) &= \left[ e^{A_0 t} + \delta A_0 t + \frac{(2A_0 \delta A + \delta A_0^2) t^2}{2!} + \dots \right] (N_0 + \delta N_0), \quad (4.4)
 \end{aligned}$$

where the subscript 0 represents values at time zero. Neglecting second-order and higher terms gives an approximate expression for the change in the evolution of nuclide densities as a function of time. The term  $\delta A_0 t$  may be factored out of the infinite sum to find a simple expression based on the unperturbed result  $N$ :

$$\begin{aligned}
N'(t) &\approx e^{A_0 t} (N_0 + \delta N_0) + \delta A_0 t N_0 + A_0 \delta A_0 t^2 N_0 + \frac{A_0^2 \delta A_0}{2!} t^3 N_0 + \dots \\
\delta N(t) &= N'(t) - N(t) \approx e^{A_0 t} \delta N_0 + \delta A_0 t N(t).
\end{aligned} \tag{4.5}$$

This result provides a First-Order Depletion Approximation (FODA) that can work with MEPT to rapidly analyze perturbations. A typical case may require 40-50 nuclides in the transmutation matrix, and cases with high spatial burn resolution spend a substantial fraction of computation on the exponentials. Detailed treatment of fission products and waste composition routinely involves treatment of over 200 nuclides, and codes do exist that explicitly model over 3000 nuclides in a full-core model. Fission product treatment is particularly important for high burnup reactors. Such large cases gain the most benefit from Eq. (4.5).

In situations where one does not wish to update the flux, Eq. (4.5) may be estimated using just the first term and neglecting the spectral  $\delta A$  term. After the modal expansion coefficients are determined and an estimate of  $\delta\phi$  is obtained, Eq. (4.5) can be recalculated with both terms, followed by a recalculation of the expansion coefficients, and so on.

#### 4.1.2 Performance of the FODA

For a zero-dimensional typical fast reactor lattice, simulations were run to compare the FODA with direct depletion calculations. Starting with a bare UZr-fueled core, a reference depletion step is computed, providing  $N_0$ ,  $N(t)$ , and  $A_0$ . Then, the

initial uranium enrichment is increased in steps up to 50% of its original value. The perturbed flux at BOC is computed for each step, providing  $\delta A_0$ . The relative errors between direct depletion and results from Eq. (4.5) for several important nuclides at each perturbation are shown in Figure 4.1. To point out the relative importance of the  $\delta N_0$  term and the spectral  $\delta A_0$  term, curves showing the relative errors for cases considering one and both terms are shown. Errors for the dominant nuclides are on the order of a hundredth of a percent for a 10% change in enrichment. For such a substantial perturbation, the results suggest that FODA is appropriate for accelerating design optimizations. The very low concentration nuclides have higher relative error, but they are present in such low quantity that the errors do not substantially affect the integral performance of the reactor.

The computational speed of FODA with just the  $\delta N_0$  term considered has been measured as over 50x faster than direct matrix-exponential depletion for a relatively small transmutation matrix (with 40 active nuclides). The avoidance of the repeated matrix-multiplications required to compute the matrix exponential accounts for this improvement, and larger depletion chains will lead to additional speedup. Considering the  $\delta A_0$  term as well reduces the relative error by more than an order of magnitude over the single-term expression and requires roughly twice the computation as the single term, leading to a 25x speedup for the 40-nuclide matrix. The substantial speed improvement of FODA comes at a steep cost in memory, as the unperturbed transmutation and burn matrices ( $A_0$  and  $e^{A_0 t}$ ) must be stored, as well as the unperturbed BOC and EOC number density vectors. However, modern computer installations enable the distribution of burn matrices across multiple compute

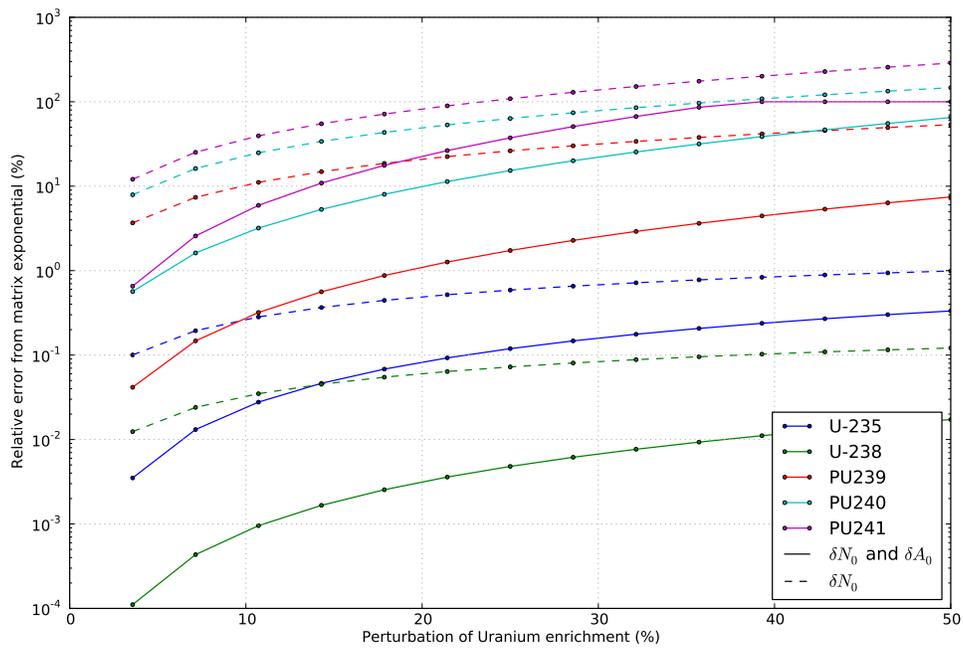


Figure 4.1: The performance of the FODA for several important nuclides, plus the effect of adding the  $\delta A$  term.

nodes that have very fast network communication, alleviating these concerns. Number density vectors can be transmitted from computer to computer very efficiently to undergo depletion through an equilibrium core simulation.

## 4.2 Applying Modal Expansion Perturbation Theory to the Equilibrium Cycle

In an equilibrium cycle, a design perturbation imposed upon the charge assemblies will result in a cascade of perturbations throughout the entire system. We would like to determine the effects of this design perturbation without directly recomputing the flux or matrix exponentials, making use of FODA and the tools developed in Chapter III.

### 4.2.1 Description of Equilibrium Cycle Perturbation Theory

A converged reference equilibrium cycle as computed by EEC described in Chapter II produces a set of converged transmutation matrices: one for each depletion subinterval of each region. Together, these  $A$  matrices provide the mapping from a freshly-charged assembly to a fully-burned discharged assembly. The action of these matrices can be modified through Eq. (4.5), considering direct  $\delta N$  terms and  $\delta A$  terms, to compute the perturbed evolution of the charge assembly. As the perturbed charge assembly follows its equilibrium path through the scheduled regions, it accumulates further nuclide density and flux perturbations until it reaches its discharge point. The process is repeated for any additional charge assemblies, depending on how many assemblies are scheduled to discharge at the end of each cycle.

Each design perturbation requires a cyclic mode iteration as described in §2.1.3.1. However, rather than relying upon direct flux calculations and depletion, ECPT implements a new region-density iteration (originally described in §2.1.3.2) that makes heavy use of MEPT and FODA. A set of design perturbations must be preceded by a reference MEPT computation with a Krylov-based solver to compute and store a number of real and adjoint flux harmonics  $\phi_i$  and  $\phi_j^*$ . At the various points in the cyclic iteration where a flux calculation is required, the current  $\delta L$  of the reactor is algebraically mapped to a global flux change  $\delta\phi$  through Eq. (3.42). This ECPT, shown as a flowchart in Figure 4.2, allows the behavior of the reactor at equilibrium to be efficiently estimated given a design perturbation at beginning-of-life.

#### 4.2.2 The Region-density iteration with FODA

Recall that cycle-average burn matrices  $B$  are required to perform the cyclic mode computation. The FODA developed in §4.1 provides a perturbation method to determine the number densities at a later time, but does not produce an explicit burn matrix. Improving the computational efficiency of the cyclic mode calculations therefore requires a new region-density iteration that is designed to take advantage of FODA.

The standard region-density iteration at a single timestep, may be graphically represented as shown in Figure 4.3. The calculation path is an iteration along the solid lines until the number densities  $N_{i+1}$  converge for each region. Full flux calculations and matrix exponential depletion calculations are used in the iteration, both of which require substantial computation. Conversely, Figure 4.4 shows the

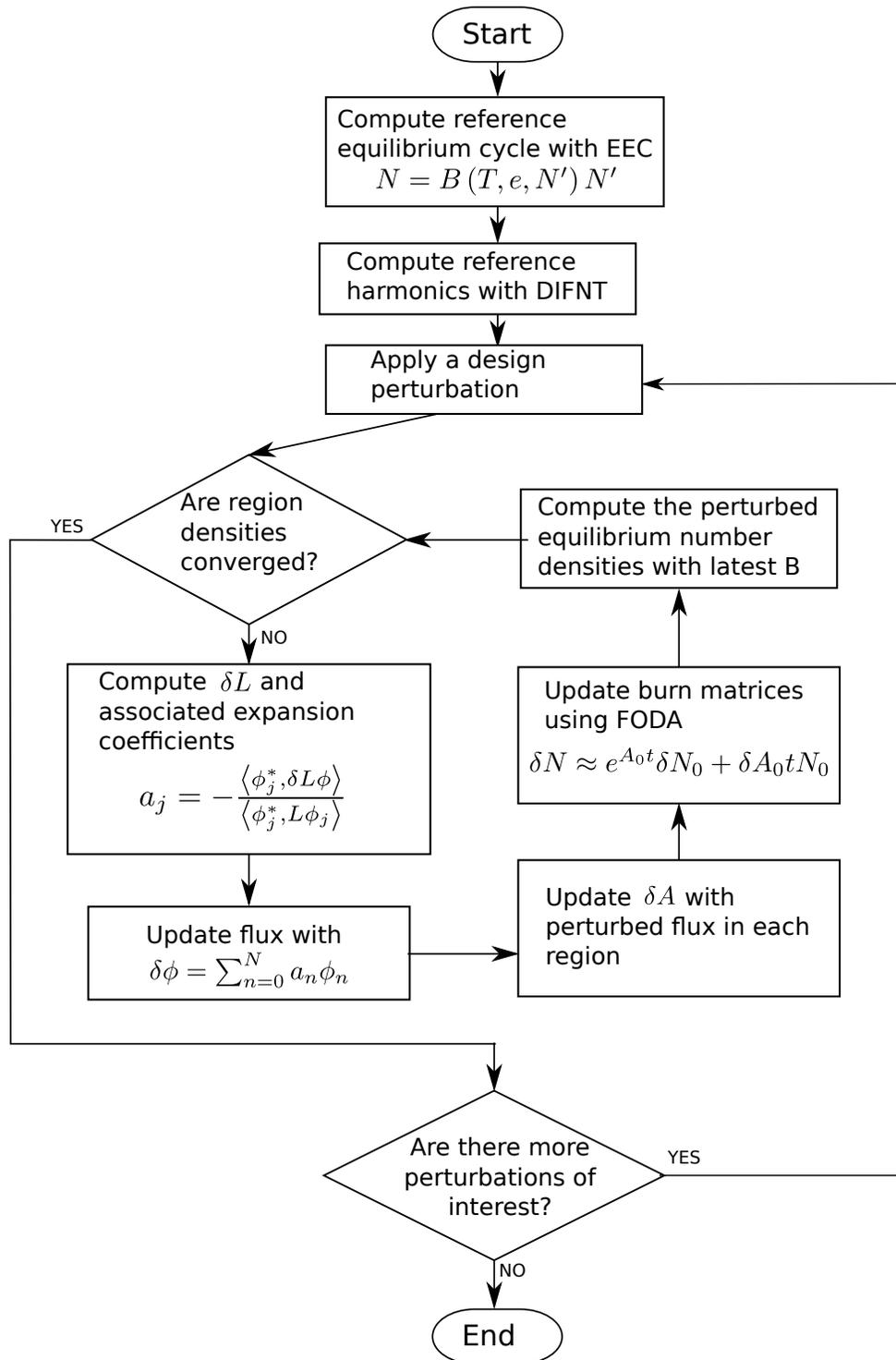


Figure 4.2: Equilibrium Cycle Perturbation Theory flowchart.

ECPT region-density iteration. From a reference state, perturbed number densities are mapped to perturbed flux with MEPT. The perturbed flux yields the  $\delta A$  terms, allowing the mapping of number densities forward in time through Eq. (4.5). The end-of-timestep densities  $N'_{i+1}$  are passed once again through MEPT to produce the flux at the next timestep. An average transmutation matrix  $\bar{A}'$  is built that subsequently produces updated number densities  $N'_{i+1}$  at each iteration. The loop continues until the number densities at timestep  $i + 1$  converge. Throughout the iteration, no direct flux calculations are required, nor are any matrix exponentials evaluated. Only after the iteration terminates, the converged-upon cycle-average transmutation matrix  $\bar{A}$  undergoes a matrix exponential to produce the burn matrix  $\bar{B}$  required by the cyclic mode iteration.

Given a reference state, the perturbation-based region-density iteration of ECPT eliminates all direct flux calculations and largely eliminates direct depletion calculations required in the evaluation of design perturbations in an equilibrium case, instead relying upon the high-performance and scalable methods developed in this thesis.

### 4.2.3 Implementation of ECPT within ARMI

ECPT has been built within the ARMI framework introduced in §2.2.1. The module follows the steps illustrated in Figure 4.2, using the EEC module with a DIF3D kernel and the internal ARMI depletion module to determine the reference solution before invoking the MEPT module with DIFNT to compute and store the reference harmonics. During the equilibrium calculation, the code is instructed to save the

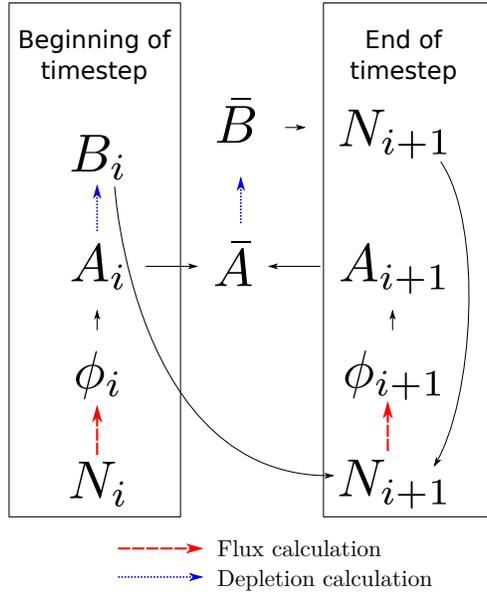


Figure 4.3: The standard region-density iteration. The cycle average burn matrix is computed by iterating along the solid lines until number densities converge.

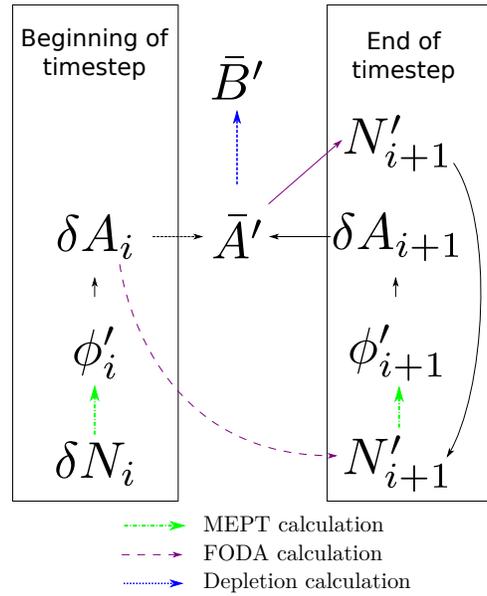


Figure 4.4: The FODA-based region-density iteration. Flux calculations and matrix exponentials are replaced by scalable, perturbation-based methods. The final burn matrix  $\bar{B}$  requires a single matrix exponential evaluation per timestep.

converged transmutation matrices for each region, where in typical equilibrium cases they would be discarded after a solution was determined. Desired perturbations are input using a small file that contains instructions to modify the charge assemblies or other aspects of the core based on a counter index that is incremented after each perturbation converges. ARMI stores pin details on the block level such that any of these are available to be perturbed, including the temperatures and dimensions of the fuel, cladding, wire wrap, coolant, and duct, as well as the isotopic makeup and material properties of any of these. Accordingly, a broad selection of perturbations are possible. The list of possible perturbations include the adjustment of the cladding thickness, duct thickness, smear density, fissile enrichment, fuel Zr fraction, thorium additive, wire wrap radius, structural impurities, and interstitial coolant space.

To enable nuclear data sensitivity analyses in equilibrium, the microscopic cross section library must be perturbed according to the proper co-variance matrices. It is certainly possible to perform these analyses within ECPT by modifying the binary ISOTXS library through ARMI interfaces rather than otherwise perturbing the reactor state. However, this research focuses on optimizations based on design perturbations rather than sensitivities to data uncertainty, which may be investigated at a later time.

### **4.3 Numerical Demonstration of ECPT**

We perform ECPT calculations for several perturbations in this section and compare to reference results computed with direct EEC. Cases are run that separate out errors due to the various levels of approximation. The reference case is that from

§3.2.3. The smear density perturbation will be focused on, as it modifies the fuel and sodium bond and is therefore a substantial perturbation. In this set, the smear density is perturbed between 65% and 68%, a range that is similar to what would be scheduled for an ECPT set during optimization.

#### 4.3.1 Perturbed flux and direct depletion

The first set of perturbations involves MEPT-based flux calculations with direct depletion. Results of three perturbations are shown in Table 4.1. As can be seen, ECPT with direct depletion agrees very well with direct EEC calculations. The reference state for this set has 66.5% smear density, but the ECPT case with 66.5% smear density does not exactly match the EEC case, nor is it expected to, as the cycle-averaging that occurs within the region density iterations uses approximate flux from MEPT at each of the intermediate time nodes rather than direct flux. Comparing the error terms in each column to the reference EEC 66.5% case shows that, for example,  $k_{eff}$  at 65% smear density is accurate to 7 pcm out of a 973 pcm change from the reference case.

The expansion coefficients for the 68% smear density case shown in Figure 4.5 demonstrate the unsurprising fact that perturbations in equilibrium cases are in general much more global than localized perturbations such as those shown in §3.3. However, while hundreds of harmonics are not necessary for this case, more than a few harmonics ( $\sim 30$ ) are still non-trivially excited.

Table 4.1: Comparison of ECPT calculations with direct depletion to EEC reference with smear density perturbations.

Smear Density	$k_{eff}$	Discharge burnup (% FIMA)	Sodium density coefficient ( $\text{¢}/^\circ C$ )	Radial expansion coefficient ( $\text{¢}/^\circ C$ )	Delayed neutron fraction ( $\beta$ )
65% (EEC)	0.9732539	31.98	0.169	-0.260	4.251E-03
65% (ECPT)	0.9719836	31.967	0.169	-0.260	4.256E-03
$\Delta^a$	<b>-131</b>	<b>-0.043%</b>	<b>-0.063%</b>	<b>0.081%</b>	<b>0.124%</b>
66.5% (EEC)	0.9827422	31.335	0.167	-0.251	4.274E-03
66.5% (ECPT)	0.9823813	31.337	0.167	-0.251	4.275E-03
$\Delta$	<b>-37</b>	<b>0.005%</b>	<b>-0.004%</b>	<b>0.026%</b>	<b>0.025%</b>
68% (EEC)	0.9918947	30.715	0.165	-0.243	4.296E-03
68% (ECPT)	0.9929172	30.735	0.165	-0.243	4.293E-03
$\Delta$	<b>103</b>	<b>0.066%</b>	<b>0.040%</b>	<b>-0.026%</b>	<b>-0.070%</b>

<sup>a</sup> $\Delta$  refers to the deviation between the ECPT values and the reference EEC values

### 4.3.2 Perturbed flux and perturbed depletion

The set of perturbations from §4.3.1 is repeated in this section with FODA activated with one and both terms. For simplicity, rather than tracking all dependent variables, only the discharge burnup is shown in these results, though other dependent variables show similar trends. Table 4.2 summarizes the results from ECPT. The results show that direct depletion is in general the most accurate, as expected. In addition, while the spectral term in FODA does reduce or maintain the error in all cases, the magnitude of reduction is much smaller than the error made between direct and single-term FODA. This suggests that single-term FODA (which is 2x faster than two-term FODA) may be sufficient for many cases, depending on the application. For scoping design optimization studies, this level of error is acceptable.

Note that in some cases (the 68% perturbation), the ECPT with direct depletion has larger error than ECPT with FODA. There is a cancellation of error in the FODA cases, where the errors in the approximate flux cancel the depletion errors in a way that manifests in lower total error. For larger perturbations, the direct depletion should always be more accurate.

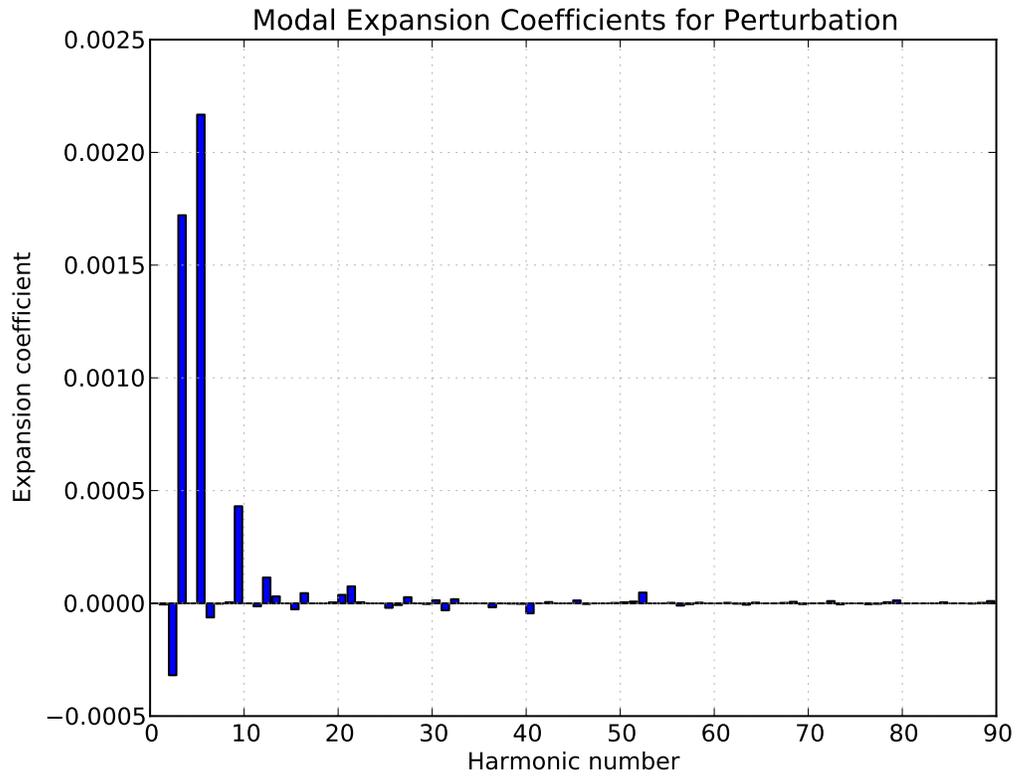


Figure 4.5: Expansion coefficients for the 68% smear density ECPT case with direct depletion.

Table 4.2: Comparison of ECPT calculations of the discharge burnup in %FIMA with FODA depletion to EEC cases with smear density perturbations.

Smear Density	Direct Depletion	FODA ( $\delta N$ term only)	FODA ( $\delta N$ and $\delta\phi$ terms)
65% (EEC)	31.980	31.975	31.975
65% (ECPT)	31.967	31.962	31.964
$\Delta^a$	<b>-0.043%</b>	<b>-0.057%</b>	<b>-0.052%</b>
66.5% (EEC)	31.335	31.327	31.327
66.5% (ECPT)	31.337	31.334	31.334
$\Delta$	<b>0.005%</b>	<b>-0.006%</b>	<b>-0.006%</b>
68% (EEC)	30.715	30.707	30.706
68% (ECPT)	30.735	30.733	30.731
$\Delta$	<b>0.066%</b>	<b>0.058%</b>	<b>0.054%</b>

---

<sup>a</sup> $\Delta$  refers to the deviation between the ECPT values and the reference EEC values

## CHAPTER V

# Multi-Objective Nuclear Reactor Design Optimization

Having extended and developed tools that can efficiently evaluate a broad set of nuclear reactor state variables and performance metrics based on design choices, we demonstrate their utility in this chapter by developing and applying a suitable multi-objective optimization scheme to assist reactor designers in finding the optimal balance between their intricately-related goals.

### 5.1 Creating a Surrogate Model of the Reactor

Although the Enhanced Equilibrium Cycle method from Chapter II and the Equilibrium Cycle Perturbation Theory method from Chapter IV are fast, users of an optimization system must be able to change the governing parameters as priorities change. Additionally, optimization algorithms evaluate their objective functions many times while traversing a design space. In complex system design, where functional partial derivatives of the objective functions are not available, the search direc-

tions are found by numerically approximating the derivatives. Practical use therefore requires nearly instantaneous evaluation of any set of design parameters. A common way to achieve such speed is by use of a surrogate model, trained to match a large set of calculations, that interpolates between the calculated points. This approach is chosen in this study.

To train such a model from a reference case, EEC or ECPT is executed for a large set of design perturbations. By virtue of their speed, a high degree of confidence in the surrogate model is assured by making many direct calculations, known as “measurements.” To enable multi-objective design optimization, the surrogate model is built as a set of multivariate regressions, one for each dependent variable, and each depending on all independent variables.

### **5.1.1 Nonlinear Regressions with the Alternating Conditional Expectation Algorithm**

Conventional linear regression models assume that the dependent variable is a linear sum of  $p$  independent variables, plus some random error. The dependencies between dependent variables and independent variables in nuclear reactor design, however, may be highly nonlinear, deeming such a model ineffective. The conventional method to fit nonlinear models is to make nonlinear transformations of the dependent and independent variables and apply linear regression to the transformed model,

$$Y = f(X_1, X_2, \dots, X_p). \quad (5.1)$$

Typical examples of the transformed approach include polynomial or logarithmic regression. This approach works well when the nonlinear transforms are known from the physics of a problem, e.g. exponential transforms for a radioactive decay problem. However, in complex multiphysics nuclear reactor simulations, the proper regression transform for many dependent variables are unknown.

The Alternating Conditional Expectation algorithm [55] has been successfully applied in similar situations, both to measured data and to model results [92, 93, 94], to determine the optimal transformations stochastically. Making no assumptions of the functional forms of the transformations  $\theta(Y)$  and  $\phi_i(X_i)$ , ACE considers the regression model as

$$\theta(Y) = \sum_{i=1}^p \phi_i(X_i), \quad (5.2)$$

and successively estimates  $\theta$  and  $\phi_i$  by iteratively minimizing the unexplained variance in the regression model,

$$\epsilon^2 = E \left\{ \left[ \theta(Y) - \sum_{i=1}^p \phi_i(X_i) \right] \right\}^2. \quad (5.3)$$

The ACE algorithm begins with arbitrary transformations. The first step of the error-minimization algorithm is to estimate each  $\phi_i$  transformation by setting it equal to the error between a given  $\theta(Y)$  and the sum of all other transformed independent function  $\phi$ , each held constant,

$$\phi_i(X_i) = E \left[ \theta(Y) - \sum_{j \neq i}^p \phi_j(X_j) \mid X_i \right]. \quad (5.4)$$

After each independent transformation is formed, the dependent transformation is updated as the solution to the minimization of Eq. (5.3), with each  $\phi$  held constant, which is

$$\theta(Y) = \frac{E[\sum_{i=1}^p \phi_i(X_i) | Y]}{\|E[\sum_{i=1}^p \phi_i(X_i) | Y]\|}. \quad (5.5)$$

Eqs. (5.4) and (5.5) form the eponymous conditional expectations of ACE. In practice, the conditional expectation values are approximated within the ACE algorithm by a local linear data smooth called the super-smoother [95].

A clear example of the capabilities of the ACE algorithm was designed by Wang [96] which shows that ACE can very successfully recover the dependent and independent functions of the expression:

$$e^y = 4 + \sin(4x_1) + |x_2| + x_3^2 + x_4^3 + x_5 + 0.1\epsilon, \quad (5.6)$$

where the random error term  $\epsilon$  makes the initial plots appear random, such that other regression models (such as ordinary least squares) would fail to recover the functional forms.

To develop the surrogate model, many ECPT cases are run in ARMI around a particular reference point. The independent variables and resulting dependent variables are saved as design perturbations are evaluated. For each dependent variable, the ACE algorithm as implemented in the ACEPACK module of R [97, 98] is executed to build the optimal transformed functions  $\theta$  and  $\phi_i$ . Cubic spline interpolations determine the value of each transformed variable for any point in design

space, as only discrete values of each transformed function are available from ACE. The dependent variable  $Y$  is determined by inverting  $\theta(Y)$  numerically, again with a cubic spline fit, to fill in the gaps between discrete values. None of the operations required during surrogate evaluation are computationally demanding, enabling very rapid execution during optimization.

ACE has an additional benefit in that plots of the transformed functions provide the designer with insight into the nature of the relationships between dependent variables and the design variables.

### **5.1.2 Training the surrogate model with ARMI**

ARMI provides a favorable infrastructure within which to build an optimization package. All modules in the system are standardized, allowing optimization routines to modify inputs such as cladding thickness so that all relevant models (including thermal, safety, and fuel performance) are affected. The automation of ARMI allows results to be determined that traditionally require substantial effort from an analyst. In particular, the ability of ARMI to evaluate a full equilibrium cycle, evaluate thermal, economic, and fuel performance, generate safety coefficients, and then run a transient code without user intervention offers an agile optimization opportunity.

The optimization module within ARMI allows users to specify an arbitrary number of degrees of freedom with respect to the independent variables. For instance, the user may request that the smear density of the fuel be varied in 10 steps while the fuel Zr-fraction be varied in 5 between specified bounds. The module submits all training cases to a computer cluster on any number of available nodes. On each

independent case, the flux harmonics of a perturbed reference case are computed with MEPT, and a set of perturbations around this reference case are evaluated with ECPT. Each perturbed reference case optionally computes a microscopic cross section library before MEPT executes. The accuracy of the results is improved if design parameters that most strongly affect the area fractions or composition of the assembly (and therefore the energy spectrum) are treated as first-level independent variables whereas others are treated as second-level independent variable, computed by MEPT. The importance of updating the cross section libraries may be minimized by using a large number of energy groups.

#### 5.1.2.1 Filtering Measurements for Regressions

Measurements from a set of ARMI runs must be conditioned before undergoing successful ACE regressions to build the surrogate model. In most practical cases considered, the cycle length is searched upon until the reactor operates in equilibrium close to a specific target  $k_{eff}$ , as described in §2.1.3.4. The criticality search introduces noise proportional to the cycle length convergence criterion. Tightening the convergence from 1000 pcm to 100 pcm dramatically improves the quality of a regression. Figure 5.1 shows a case with a 100 pcm convergence criterion on the cycle length, and Figure 5.2 shows the same case with a 1000 pcm convergence. The regression quality for the tightly-converged case is substantially improved, but does not improve further with tighter convergence.

When the measurements are stratified evenly throughout all dimensions, certain regions will not have a physically-meaningful solution (e.g. when the reactor is

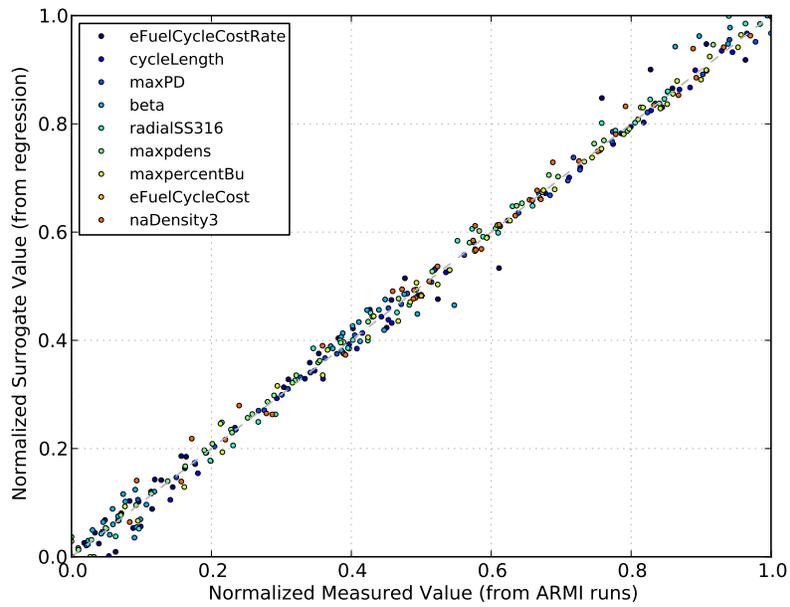


Figure 5.1: Sample regression quality for a critical case with a 100 pcm convergence criterion.

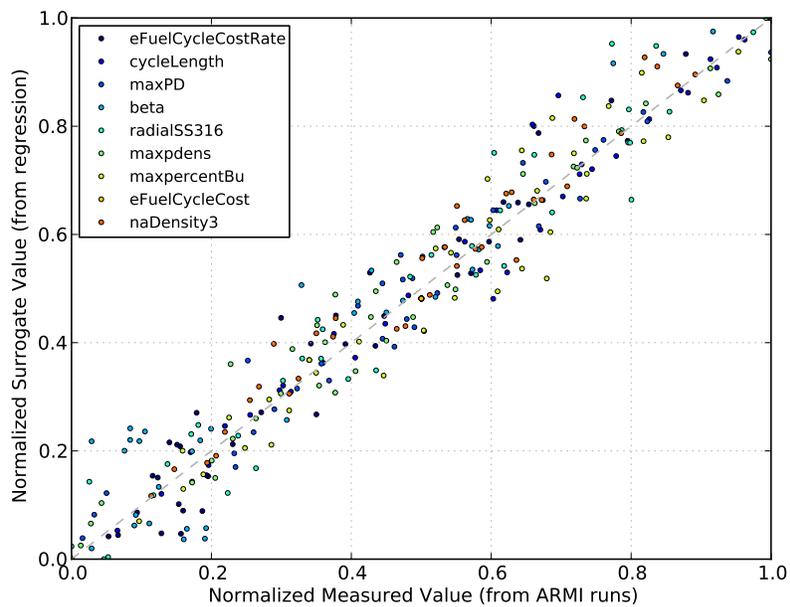


Figure 5.2: Regression quality for a critical case with a 1000 pcm convergence criterion.

subcritical or supercritical at all cycle lengths). To avoid degradation, the regression routine must reject points that do not satisfy the convergence criterion.

Criticality typically follows a parabolic dependence on cycle length, with up to two critical roots. As such, the critical cycle length may switch from the low point to the high point when the first root cycle length becomes negative. This behavior can result in a step-increase in cycle length of 1000 days or more, and therefore adds a substantial jump in all dependent variables which is challenging to match with a regression. It is therefore essential to monitor the sign of the derivative of the criticality curve vs. cycle length and detect such jumps, eliminating those with the less prevalent sign.

In cases that contain a criticality search, all dependent variables must be measured at BOEC. In such cases, the cycle length varies as a complicated function of all independent variables, and the ACE algorithm does not perform well if EOEC values are measured. If the determined cycle length is treated as an independent variable, the ACE algorithm performs the regression well at EOEC. However, this regression is not meaningful unless the functional dependence of the cycle length on the true independent variables is available. A two-level regression is possible using this concept, and may be investigated in later studies.

### **5.1.3 Sample Surrogate of a High-Burnup Fast Reactor**

We demonstrate the use of ACE for creating a surrogate model of the reactor in this section. The reactor model is identical to that from §3.2.3. Training occurs with three independent variables, the charge enrichment, the cladding thickness, and the

Table 5.1: The 125-point test matrix for the sample surrogate model.

Independent Variable	Minimum	Maximum	Number of Points
Charge Enrichment (wt. %)	10.0	12.0	5
Cladding Thickness (mm)	2.0	6.0	5
Smear Density (%)	55	70	5

smear density of the fuel. The test matrix is evenly stratified according to Table 5.1. The ARMI code processes the test matrix and builds the 125 cases from an initial reference case, applying each combination of independent variables to the core and then running EEC calculations to reach an equilibrium condition. Once equilibrium is reached, reactivity coefficients are computed. Dependent variables tracked in this case include: critical cycle length, maximum discharge burnup, maximum volumetric power density, fuel cycle cost, delayed neutron fraction  $\beta$ , radial expansion coefficient of reactivity, and sodium density coefficient of reactivity.

Of the 125 points executed, regression filtering eliminated 80 of them, leaving 40 measurements for the regression. A higher acceptance rate may be achieved by narrowing the test matrix in Table 5.1 to have smaller ranges. If the acceptance rate is too low to produce a regression for a specific test matrix, an additional set of measurements must be run in an iterative process that may be automated.

The sample case requires 7x three-variate ACE regressions; one for each dependent variable. To determine the quality of the regression, exact measured values are compared to the values given by the surrogate model for the specific independent variable values that were used in the training. A normalized plot of the regression value vs. the measured value for all independent variables for this case was shown previously in Figure 5.1. The predictive quality of the regression is tested by train-

Table 5.2: Standard deviation in the fractional error between measured points and their surrogate values for all points and for a predictive 90% model. The average total error has been accounted for in the  $\theta(Y)$  transform. The standard deviations for the predictive case are lower because only 10% of the measured points were compared vs. 100% in the first column.

Dependent Variable	All Measurements	90% of Measurements
Critical Cycle Length	0.0095	0.0039
Discharge Burnup	0.0058	0.0026
Power Density	0.0011	0.0003
Fuel Cycle Cost	0.0085	0.0027
Delayed Neutron Frac.	0.0024	0.0006
Radial Expansion Coef.	0.0026	0.0007
Sodium Density Coef.	0.0096	0.0036

ing the model on only 90% of the measurements, comparing the remaining 10% of measurements against their resulting regression. The predictive qualities of the surrogate from all points and 90% of points are quantified as the standard deviation of fractional errors in Table 5.2. The variance for the predictive model are below 1% in general, suggesting that the interpolative quality of the surrogate is high, and therefore that the surrogate is satisfactory for use in design optimization within the range of the test matrix.

As mentioned previously, the ACE regressions not only provide a high-quality surrogate model, but also provide designers with meaningful insight into the relations between the independent and dependent variables. For example, the dependent and independent transforms  $\phi(X)$  and  $\theta(Y)$  shown in Figure 5.3 may be interpreted as follows. Since the dependent transform is monotonically increasing, higher values of the independent transforms will result in higher discharge burnup. With this in mind, it is clear that minimizing the smear density, maximizing the cladding

thickness, and minimizing the charge enrichment will result in the lowest discharge burnup. Furthermore, considering the values of the independent transforms along the y-axes, it can be seen that the discharge burnup is an order of magnitude more sensitive to the smear density than to the other two independent variables over their tested ranges.

Note that these trends to minimize burnup all reduce the reactivity of the fuel, thus requiring a shorter cycle length to maintain criticality as the enriched fuel burns out. The transforms for the cycle length dependent variable intuitively exhibit identical trends to those for discharge burnup. In depleted- or natural-uranium fed TWRs, the trends for the same goal would be substantially different.

The number of points in each independent transform  $\phi(X)$  show number of test-matrix points that passed regression filtering, thereby determining the tighter range that may be re-trained for a finer second pass. Additionally, the abscissa of the transform  $\theta(Y)$  shows the breadth of the test matrix for each dependent variable. The discharge burnup in Figure 5.3 ranges from 25 to 38 % FIMA, and is thus a broad test matrix, covering a wide range of design performance, even though 2/5 of the smear density points and 1/5 of the charge enrichment points were eliminated by filtration.

More nonlinear behavior between independent and dependent variables can be seen in the transforms for equilibrium fuel cycle cost rate shown in Figure 5.4, which considers the costs of feed material, enrichment, and fabrication. The transform of the charge enrichment shows the ability of the cubic splines in fitting nonlinear trends. More importantly, the essential point for a reactor designer to gather

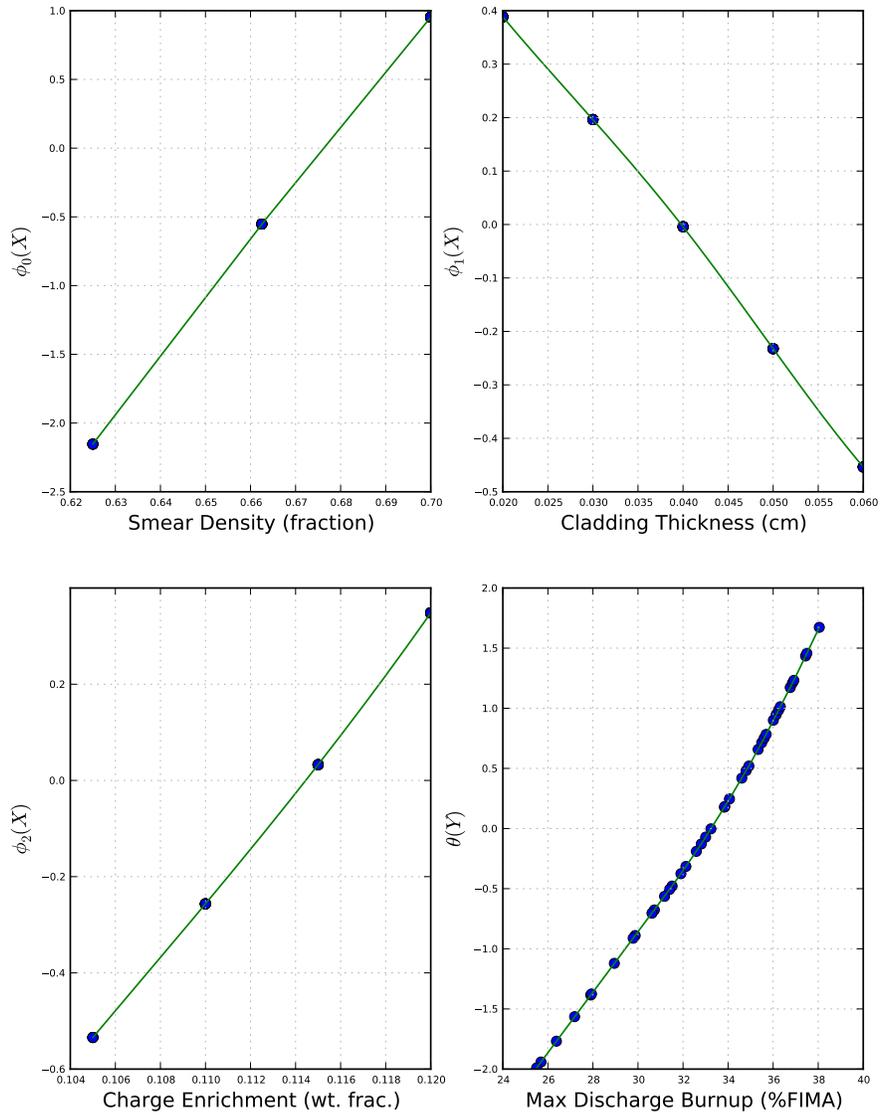


Figure 5.3: The ACE transforms for the maximum discharge burnup. The first three plots represent transforms of the independent variables, and the fourth shows the transform of the dependent variable.

from this regression data is that the fuel cycle cost is nearly invariant to the charge enrichment. On deeper investigation, the designer will discover that while higher enrichment feed assemblies cost more to mine, enrich, and fabricate, their additional reactivity allows the cycle length to be extended, therefore reducing the frequency with which assemblies must be manufactured. That this counterintuitive fact is automatically extracted from hard-to-visualize multidimensional set of experiments clearly demonstrates the conceptual utility of such a regression model.

While ACE surrogate model shows the relations between the model inputs and outputs, cases with many objectives to optimize concurrently still present the reactor designer with a challenge: how to balance the various objectives against each other. This is the focus of the field of multi-objective optimization.

## **5.2 Multi-Objective Optimization with Physical Programming**

Given a high-quality surrogate of multidimensional design space, the final step towards automated optimization involves locating the design parameters that meet the designer's multidisciplinary goals and constraints. Before discussing our approach, some basic concepts of multi-objective optimization are reviewed.

### **5.2.1 Multi-Objective Optimization Background**

multi-objective optimization (MOO) problems are framed as follows [99]:

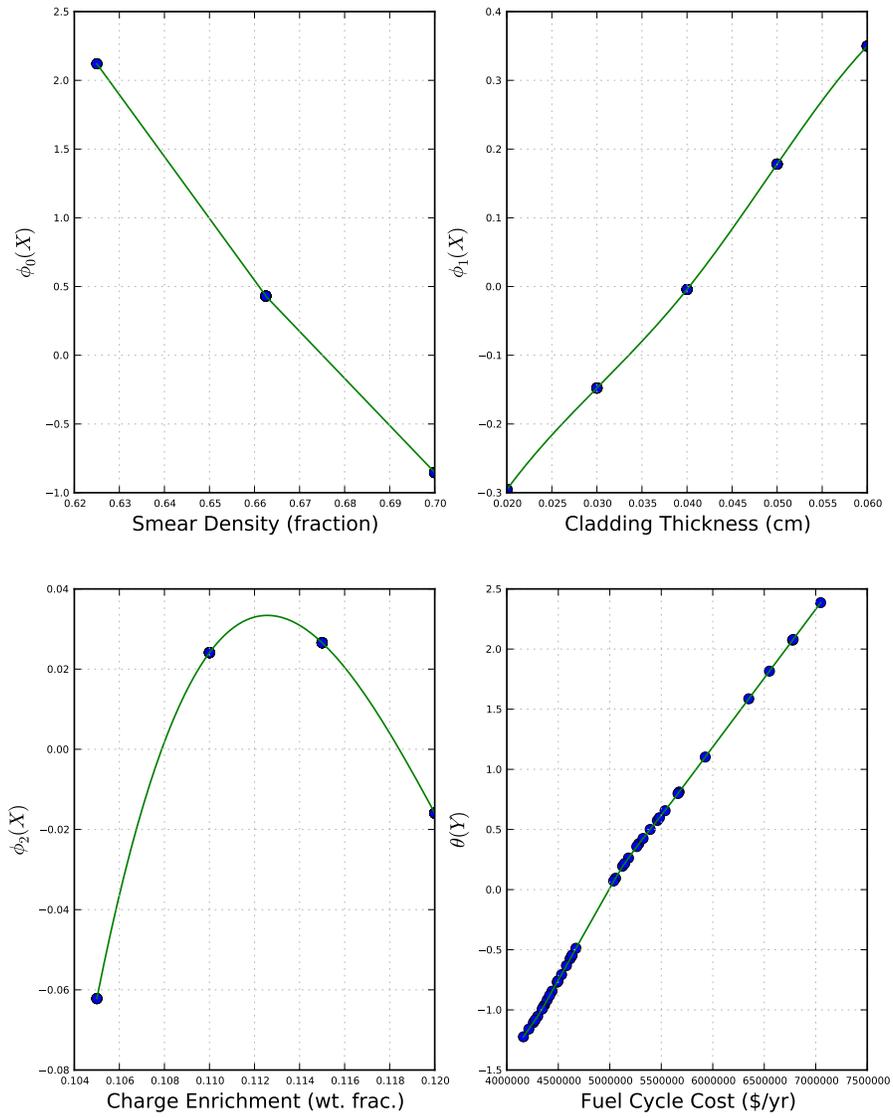


Figure 5.4: The ACE transforms for the equilibrium fuel cycle cost rate.

$$\begin{aligned}
& \underset{x}{\text{minimize}} && F(x) = [F_1(x), F_2(x), \dots, F_k(x)]^T \\
& \text{subject to} && g_j(x) \leq 0, \quad j = 1, 2, \dots, m, \\
& && h_l(x) = 0, \quad l = 1, 2, \dots, e,
\end{aligned}$$

where  $k$  is the number of objective functions,  $m$  is the number of inequality constraints, and  $e$  is the number of equality constraints.  $x$  is a vector of design variables, and has length equal to the number of independent variables.

Single-objective optimization problems have the straightforward goal of minimizing or maximizing a single objective, subject to constraints. When there are multiple objectives, the concept of “optimal” is less obvious. Rather than locating a single optimal point, multi-objective problems are solved by finding sets of points that are known as *Pareto optimal*. A point  $x^*$  is Pareto optimal if and only if there does not exist another point,  $x \in X$ , such that  $F(x) \leq F(x^*)$ , and  $F_i(x) < F_i(x^*)$  for at least one other function. In other words, a point is Pareto optimal if there is no other point that improves at least one objective function without diminishing any of the others.

Weak Pareto Optimality is a similarly useful concept, as non-Pareto optimal points are often still of practical use. A point  $x^*$  is weakly Pareto optimal iff there does not exist another point  $x \in X$  such that  $F(x) < F(x^*)$ , i.e., there is no other point which improves all the objective functions. Weak Pareto optimality is a necessary condition for Pareto optimality.

Many algorithms exist to locate Pareto optimal points. The most common fall into the category of global criterion methods, which weight the  $F_k$  objective functions with some a priori preferences to obtain a single objective function to optimize.

### 5.2.2 Introduction to Physical Programming

The fundamental phases of design optimization are as follows [100]:

1. Choose design parameters: determine which aspects of design are independent variables. For high-burnup fast reactors, these might include smear density, Zr %, enrichment, fuel management, clad thickness, duct thickness, interstitial gap thickness, cycle length, total power, and others.
2. Choose design metrics: what results do we care about? These are the dependent variables. Examples may include discharge burnup, discharge DPA, equilibrium  $k_{eff}$ , reactivity coefficients, cladding wastage, peak cladding temperature in an unprotected loss-of-flow accident, fuel cost, etc.
3. Develop a mapping between design parameters and design metrics: given a set of design parameters, what are the resulting design metrics? This is the computationally-intensive step of running the nuclear reactor simulations. ARMI is built to provide this mapping.
4. Develop an aggregate objective function: to reflect design preference, each of the design metrics are combined to form a single expression, that, if minimized, will satisfy the designer's preferences.

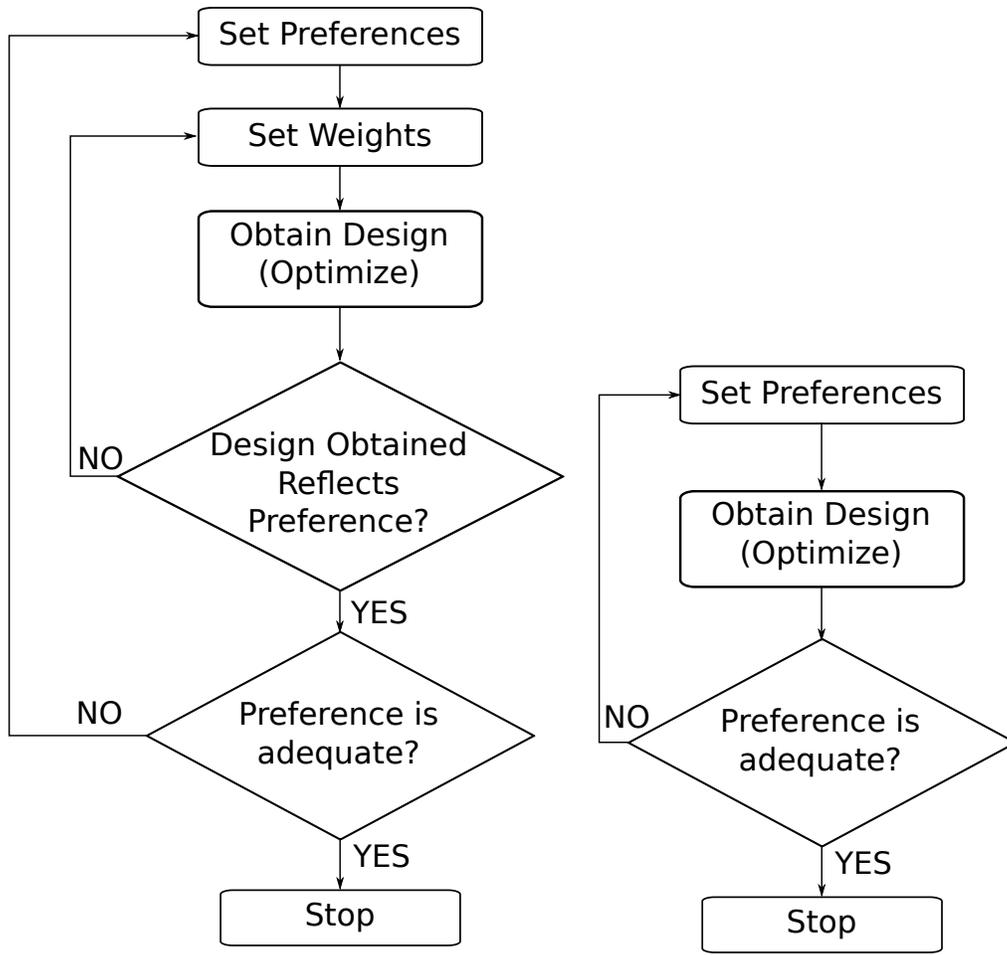
5. Perform computational optimization: use any number of well-known optimization codes to minimize the aggregate objective function subject to constraints.

Out of these steps, step (4) is perhaps the least straightforward in many applications, including nuclear reactor optimization. The simplest approach is to build a weighted sum of the various design metrics. In this approach, the weights do not have strong physical analogies, and the process of determining a meaningful set is iterative and challenging.

Physical Programming (PP) was designed to bypass the imprecise art of determining weighting functions as shown schematically in Figure 5.5, instead allowing users to directly build a single set of physically-meaningful preferences [56, 100]. In the PP approach, preferences are specified based on the the four class functions shown in Figure 5.6. Each class has a soft case, which will become a component of the aggregate objective to minimize, and a hard case, which will be used as a constraint in the optimization routines. Class functions transform the values of each dependent variable  $g_i$  into strictly-positive, convex, and unitless values  $\bar{g}_i$ , such that minimizing the aggregate sum of class functions is always akin to approaching the desired range in each of the objectives. PP provides a system to automatically generate the weights and normalizations of an aggregate objective based on a straightforward preference selection.

The explicit shape of the functions (shown in Figure 5.7) is informed by Physical Programming lexicon, which defines ranges for each dependent variable  $i$  as shown in Table 5.3.

Users choose physically-meaningful values as the class parameters  $g_{i1}$  through  $g_{i5}$



a) Weight-based approach

b) Physical programming approach

Figure 5.5: Flow charts showing the advantage of the physical programming approach over traditional weighting methods.

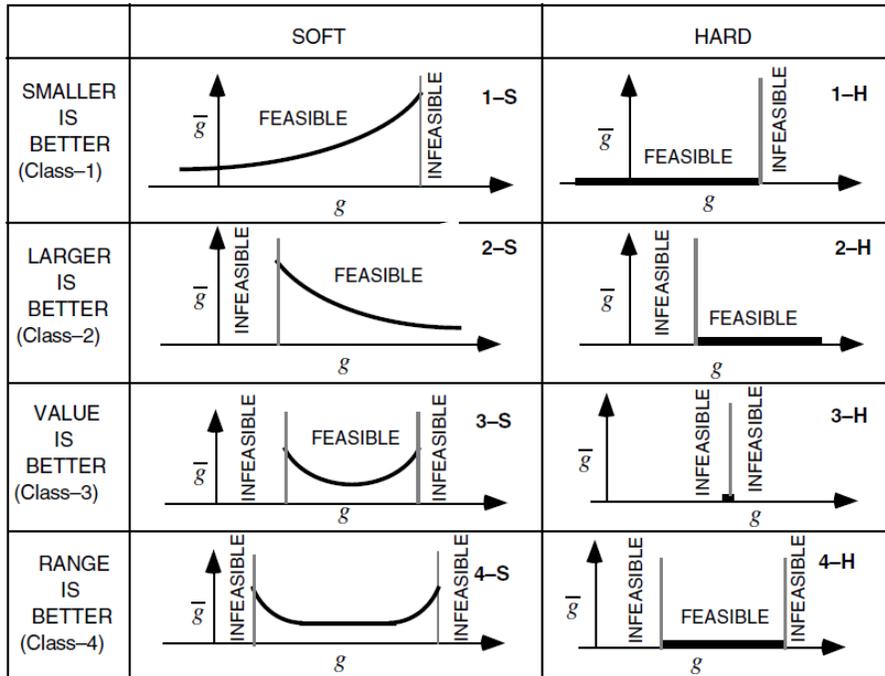


Figure 5.6: General shapes of the class functions (from [100]).

Table 5.3: The class parameters defined in Physical Programming lexicon.

Highly Desirable range	$g_i \leq g_{i1}$
Desirable range	$g_{i1} \leq g_i \leq g_{i2}$
Tolerable range	$g_{i2} \leq g_i \leq g_{i3}$
Undesirable range	$g_{i3} \leq g_i \leq g_{i4}$
Highly Undesirable range	$g_{i4} \leq g_i \leq g_{i5}$
Unacceptable range	$g_i \geq g_{i5}$

for each dependent variable and constraint according to their understanding of the problem. Explicit class functions that exhibit several convenient mathematical properties may be built based on these parameters, typically as piecewise splines. Once explicit class functions are defined, any off-the-shelf optimization method may be executed to traverse the multi-dimensional curve, searching for the global optimum.

### 5.2.3 Mathematical Construction of the Class Functions

The class functions must satisfy certain mathematical properties, elaborated in [56], to allow the effective application of convex optimization algorithms. To avoid introducing local minima, they must be strictly positive, have a continuous first derivative, and have a strictly positive second derivative for all possible sets of preference boundaries. Additionally, the class functions should follow a one-vs.-others rule, which dictates, for example, that a single outlying criterion should be brought into its tolerable range before all others are brought from tolerable to desirable.

Following Messac, one approach to building the class functions involves using 4-th order splines. The form of spline segments that have suitable class function properties can be determined by forcing the second derivative of each class function  $\bar{g}(g)$  in each preference region  $k$  to be positive. We begin searching for a suitable spline function by choosing a second derivative expression that we can easily force to be positive and integrating it until the spline expression is found. We choose the second derivative as:

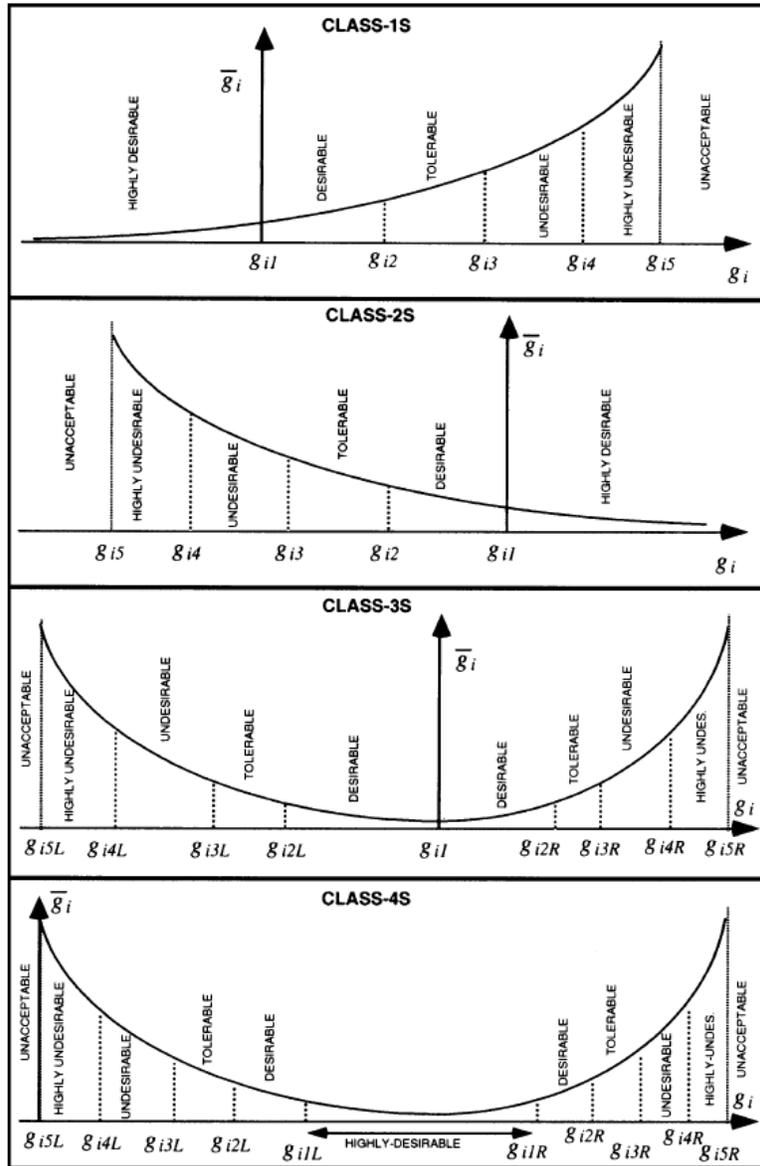


Figure 5.7: Physically-meaningful class function definitions (from [100]).

$$\frac{\partial^2 \bar{g}}{\partial g^2} \equiv \lambda_k^2 [a (\xi_k)^2 + b (\xi_k - 1)^2], \quad 0 \leq \xi_k \leq 1 \quad (5.7)$$

where  $a$  and  $b$  are real constants,  $\xi_k$  is the fraction between endpoints  $g_{k-1}$  and  $g_k$ , defined as

$$\xi_k = \frac{g - g_{k-1}}{g_k - g_{k-1}}, \quad (5.8)$$

and where  $\lambda_k$  is the distance between preference region  $k$  endpoints,

$$\lambda_k = g_k - g_{k-1}. \quad (5.9)$$

If we choose constants  $a$  and  $b$  such that they are strictly positive, we are guaranteed that the second derivative of the spline segment is also positive, as desired. Integrating with respect to  $g$ , the first derivative is:

$$\frac{\partial \bar{g}}{\partial g} = \lambda_k^3 \left\{ \frac{a}{3} [\xi_k(g)]^3 + \frac{b}{3} [\xi_k(g) - 1]^3 \right\} + c. \quad (5.10)$$

A second integration gives the functional form of the class function  $\bar{g}(g)$  as:

$$\bar{g}(g) = \lambda_k^4 \left\{ \frac{a}{12} [\xi_k(g)]^4 + \frac{b}{12} [\xi_k(g) - 1]^4 \right\} + c\lambda_k \xi_k(g) + d. \quad (5.11)$$

In these integrations, the class function  $\bar{g}(g)$  depends on the dependent variable  $g$  through the auxiliary function  $\xi(g)$ , and the  $\lambda_k$  terms increase in power when integrated due to the chain rule. To ensure that the class functions are convex, conditions under which the constants  $a$  and  $b$  are positive by Eq. (5.7) are determined

by imposing arbitrary points  $(g_{k-1}, \bar{g}_{k-1})$  and  $(g_k, \bar{g}_k)$  and slopes  $s_{k-1}$  and  $s_k$  on the region boundaries at  $\xi_k = 0$  and  $\xi_k = 1$  and then finding values that suit positivity.

From Eq. (5.11),

$$\bar{g}_{k-1} = \lambda_k^4 \left( \frac{b}{12} \xi^4 \right) + d, \quad (5.12)$$

$$\bar{g}_k = \lambda_k^4 \left( \frac{a}{12} \xi^4 \right) + c\lambda_k + d, \quad (5.13)$$

and from Eq. (5.10),

$$\left. \frac{\partial \bar{g}}{\partial g} \right|_{g_{k-1}} = s_{k-1} = -\lambda_k^3 \left( \frac{b}{3} \right) + c, \quad (5.14)$$

$$\left. \frac{\partial \bar{g}}{\partial g} \right|_{g_k} = s_k = \lambda_k^3 \left( \frac{a}{3} \right) + c. \quad (5.15)$$

These four equations can be solved simultaneously to determine expression for the four constants, finding

$$a = \frac{3(s_k + s_{k-1}) - 12\tilde{s}_k}{2\lambda_k^3}, \quad (5.16)$$

$$b = \frac{12\tilde{s}_k - 3(s_k + 3s_{k-1})}{2\lambda_k^3}, \quad (5.17)$$

$$c = 2\tilde{s} - \frac{1}{2}s_k - \frac{1}{2}s_{k-1}, \quad (5.18)$$

$$d = \bar{g}_{k-1} - \lambda \left( \frac{1}{2} \tilde{s} - \frac{1}{8} s_k - \frac{3}{8} s_{k-1} \right), \quad (5.19)$$

where  $\tilde{s}_k = \frac{\Delta \bar{g}_k}{\lambda_k}$  is a characteristic slope of region  $k$ , and  $\Delta \bar{g}_k$  is the change in  $\bar{g}$  encountered when crossing region  $k$ . The values of  $\Delta \bar{g}_k$  that enabled positivity of the second derivative are unknown a priori, and must be determined through an iteration. The slopes and  $\bar{g}_k$  values for a given set of  $\Delta \bar{g}_k$  values are computed iteratively until both  $a$  and  $b$  are positive, ensuring that the class functions are convex. The values of  $\Delta \bar{g}_k$  are identical for each objective by design, such that crossing the tolerable region, for example, is of equivalent worth between each objective.

Imposing positivity inequalities on the Eqs. (5.16) and (5.17), we find required bounds for the slopes of the curves on each side of region  $k$ :

$$4\bar{s}_k - 3s_k \leq s_{k-1} \leq \frac{4\bar{s}_k - s_k}{3} \quad (5.20)$$

$$\frac{4\bar{s}_k - s_{k-1}}{3} \leq s_k \leq 4\bar{s}_k - 3s_{k-1}. \quad (5.21)$$

These bounds suggest the following algorithm to determine which slopes and class function values should be imposed to build the splines. Starting with a highly desirable region height of 0.1, the height of the second region of a problem with  $n_{sc}$  soft criteria is set to  $\beta n_{sc} \Delta g_1$ , where  $\beta$  is a convexity parameter. In this way, the height for region 2 is specifically designed to follow the one-vs.-others rule. The slope at the boundary between region 1 and region 2 is set to be between the bounds determined by Eq. (5.21) according to a parameter  $\alpha$ . If  $\alpha = 0$ , the lower bound of the slope is

chosen, whereas  $\alpha = 1$  leads to the upper bound. Once a slope is chosen, the values for  $a$  and  $b$  are computed with Eqs. (5.16) and (5.17) to ensure their positivity and thus the convexity of the spline segment. The process is repeated for all regions, as shown as Algorithm (V.1), and is run simultaneously for all objectives until a  $\beta$  value is found that satisfies convexity for each. The accepted values are finally used to compute numerical values for the constants  $a$ ,  $b$ ,  $c$ , and  $d$ , completing the generation of the spline-based class functions.

---

**Algorithm V.1** Determining the slopes and values of the class functions.

---

```

Set  $\bar{g}_1 = \Delta g_1 = 0.1$ 
Set  $\beta = 1.5$ 
loop
  for  $k = 2, 3, 4, 5$ 
    set  $\Delta g_k = \beta n_{sc} \Delta g_{k-1}$ 
    set  $\bar{g}_k = \bar{g}^{k-1} + \Delta g_k$ 
    set  $\lambda_k = g_k - g_{k-1}$ 
    set  $\tilde{s}_k = \frac{\Delta g_k}{\lambda_k}$ 
    set  $s_k = \frac{4\tilde{s}_k - s_{k-1}}{3} + \alpha \left[ \frac{8}{3} (\tilde{s}_k - s_{k-1}) \right]$ 
  if  $a > 0$  and  $b > 0$  for each region
    done
  else
    set  $\beta = \beta + 0.5$ 
    continue loop

```

---

Example class functions from a 2-objective problem determined by this algorithm are shown in Figures 5.8 and 5.9. They represent an arbitrary set of preferences on

two relevant dependent variables, the void worth and the volumetric power density. Note that the vertical change between region boundaries is identical for the two objectives, even though the values of the dependent variables are two orders of magnitude apart.

#### 5.2.4 Minimizing The Aggregate Objective Function

Having built class functions for each objective that map our dependent variables to normalized strictly-positive convex functions, the final step of optimization is minimizing an aggregate objective function. Following Messac [56], we choose

$$g = \log_{10} \left( \frac{1}{n_{sc}} \sum_{i=1}^{n_{sc}} \bar{g}_i \right), \quad (5.22)$$

where  $\bar{g}_i$  is the spline function defined by Eq. (5.11). The logarithm transforms the class functions such that they typically span only a few orders of magnitude, a favorable condition for nonlinear programming algorithms. This function can be minimized with most standard quasi-Newton or otherwise gradient-based optimization functions such as Newton Conjugate Gradients [101], Limited-memory Boyden-Fletcher-Goldfarb-Shanno Bounded (L-BFGS-B) [102, 103], Constrained Optimization BY Linear Approximation (COBYLA) [104], Sequential Least Squares Programming (SLSQP) [105], etc. Global optimization algorithms such as simulated annealing [106] and brute force can also be applied, but will generally perform more slowly than the gradient-based method since the problem has been formed as convex. The optimization statement is

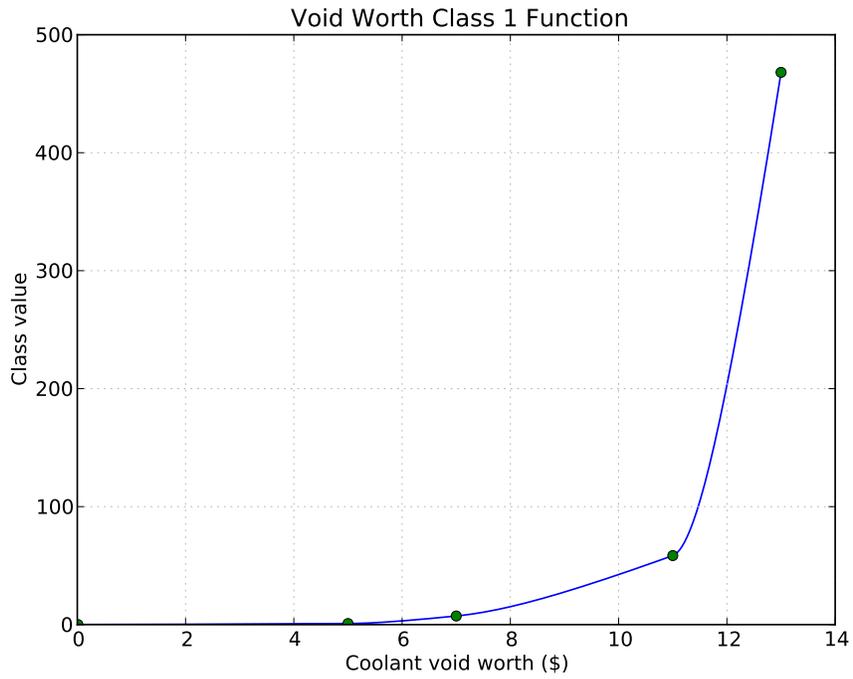


Figure 5.8: An example Class 1 “smaller is better” function.

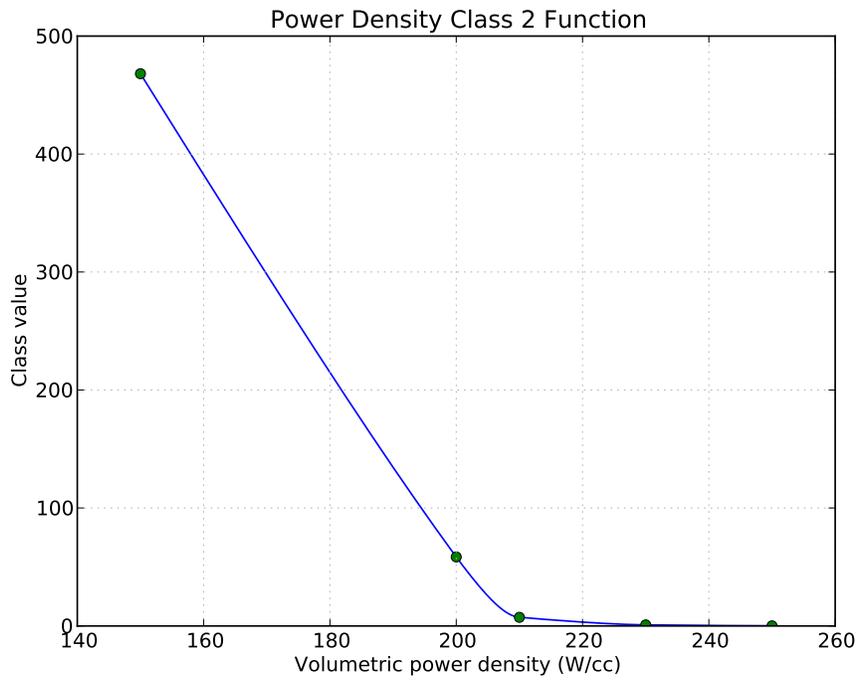


Figure 5.9: An example Class 2 “larger is better” function. Note that the class values at each point are identical to those in Figure 5.8.

$$\begin{aligned}
& \underset{x}{\text{minimize}} && g(x) = \log_{10} \left( \frac{1}{n_{sc}} \sum_{i=1}^{n_{sc}} \bar{g}_i(g_i(x)) \right) \\
& \text{subject to} && g_i(x) \leq g_{i5}, \quad \text{class 1-S,} \\
& && g_i(x) \geq g_{i5}, \quad \text{class 2-S,} \\
& && g_{i5L}(x) \leq g_i(\mathbf{x}) \leq g_{i5R}, \quad \text{class 3-S,} \\
& && g_i(x) \leq g_{iM}, \quad \text{class 1-H,} \\
& && g_i(x) \geq g_{im}, \quad \text{class 2-H,} \\
& && g_{im} \leq g_i(x) \leq g_{iM}, \quad \text{class 3-H,}
\end{aligned}$$

where  $g_{im}$  and  $g_{iM}$  represent minimum and maximum values. This form of the optimization statement is readily applied to most optimization routines.

### 5.2.5 Sample Optimization of a Simple Nuclear Reactor Model

We continue the treatment of the example case from §3.2.3 and §5.1.3 by performing PP-based optimization on the problem with the direct solver. Four dependent variables are chosen to drive the optimization, with PP region boundaries defined as shown in Table 5.4.

The boundaries are chosen based upon physically-meaningful values. Ideally, the sodium density coefficient would be below zero, but large, low-leakage fast reactors cannot typically achieve this. However, transient performance with coefficients as high as 0.2  $\text{¢}/K$  have achieved inherent safety. Above 0.35, inherent safety and reactor stability become questionable. Discharge burnup on the order of 25% FIMA

has been achieved in fast reactors, but once-through performance is expected to improve with burnup. However, the farther away the discharge becomes from the known database, the longer the fuels and materials test program must be to qualify such a fuel. Therefore, burnup should be minimized. The burnup maximum is set by the fact that fission products in a metal-fueled sodium-cooled spectrum begin to poison the reactor above 40% FIMA. Power density should be maximized to improve economics, since a plant that can achieve a power uprate without increasing its physical size can reduce the cost of electricity. Finally, the fuel cycle costs should be minimized.

The four class functions for the dependent variables in Table 5.4 are computed with Algorithm V.1. These are applied to the surrogate model from §5.1.3 to form the aggregate objective from Equation 5.22. Hard constraints for each independent variable are inferred from the test matrix in Table 5.1 and from the regression filtration process, such that the bounds of the aggregate objective function correspond with the bounds displayed along the abscissas of Figure 5.3. Several constrained optimization algorithms are executed on the case, with results shown in Table 5.5.

All algorithms converged towards the same design, with slight exception from the COBYLA algorithm, which reduced the cladding thickness below the others. The agreement with the brute force algorithm, which is global in scope, demonstrates that the optimal design is indeed a global optimum in the design variable range. In all cases, the PP regions for each dependent variable are identical: the sodium density coefficient is Tolerable, the discharge burnup is Tolerable, the power density is Desirable, and the fuel cycle cost is Undesirable.

Table 5.4: Physical Programming region boundaries for sample problem

Dependent Variable	Class	Highly Desirable	Desirable	Tolerable	Undesirable	Highly Undesirable
Sodium Density Coeff. ( $\text{¢}/^\circ\text{C}$ )	S	0.01	0.1	0.2	0.3	0.35
Max. Discharge Burnup (% FIMA)	S	25	28	31	35	40
Max. Power Density ( $\text{W}/\text{cm}^3$ )	L	250	130	120	110	100
Fuel Cycle Cost ( $\text{\$ M}/\text{yr}$ )	S	4.0	4.5	5.0	7.0	10.0

Table 5.5: Optimization results from various optimization algorithms

Algorithm	Charge Enrich. (wt. %)	Cladding Thickness (mm)	Smear Density (%)	$\alpha_{Na}$ ( $\text{¢}/^\circ\text{C}$ )	Max. BU (% FIMA)	Power Density ( $\text{W}/\text{cm}^3$ )	Fuel Cycle Cost ( $\text{\$ M}/\text{yr}$ )	Objective Minimum
L-BFGS-B	10.5	0.600	67.9	0.171	29.9	149.1	5.11	1.06382
Newton CG	10.5	0.600	67.9	0.171	29.9	149.2	5.12	1.06369
COBYLA	10.5	0.523	67.9	0.178	30.6	147.0	5.02	1.07961
SQLSP	10.5	0.600	67.9	0.171	29.9	149.2	5.12	1.06369
Brute Force	10.5	0.600	67.9	0.171	30.0	148.7	5.10	1.064108

Table 5.6: The 125-point test matrix for the sample TWR optimization model. All points were executed with ECPT using a total of 25 reference points.

Independent Variable	Minimum	Maximum	Number of Points
Charge Enrichment (wt. %)	0.3%	5.0%	5
Cladding Thickness (mm)	0.2	0.6	5
Smear Density (%)	55	70	5

### 5.3 Optimization of a High Burnup Fast Reactor

With the final set of tools developed, we demonstrate their combined capabilities in this section by optimizing a high-burnup fast reactor design. This case includes a realistic core layout with axial and radial shields, a gas plenum, control rod positions, and has 5x more mesh points than the simple optimization from §5.2.5. Burnup dependent cross sections and fuel performance coupling are active, and ECPT is used to compute all of the data points in place of the direct solver. The core map for the case is shown in Figure 5.10. In this case, a criticality search is not performed.

Recall from §1.2 that the primary goal of the TWR is to reach high burnup while minimizing enrichment and reprocessing. The capability of some designs to operate using depleted uranium as fuel enables equilibrium operation without any continued enrichment. While this feature is highly desirable, the possibility of feeding low-enriched uranium warrants study as well. Therefore, we study the effects of varying the charge enrichment up to 5%, which is on the order used in traditional LWRs. We also vary the cladding thickness and smear density as in the previous sample case. The test matrix is described in Table 5.6.

The reference harmonics are computed at equilibrium, which is slightly asymmetric, and are therefore themselves asymmetric. The midplane of the 40th adjoint

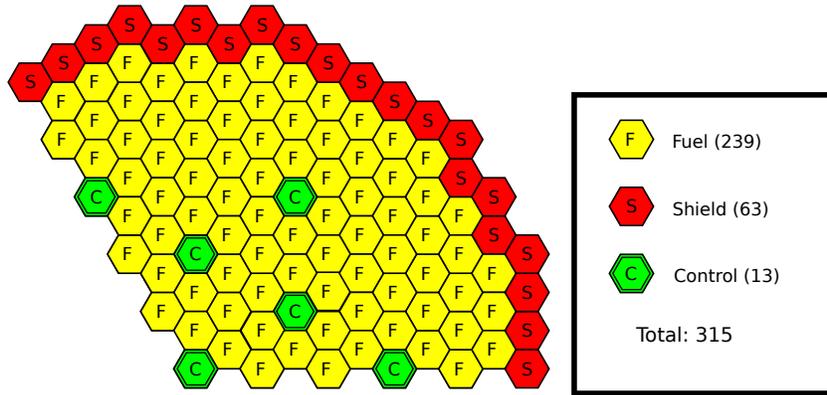


Figure 5.10: The core map for the high burnup fast reactor optimization case.

harmonic is shown in Figure (5.11), where the localized nature of the higher harmonics is evident.

The initial optimization of this case is performed with the preferences shown in Table 5.7. Performance metrics monitoring the maximum discharge cladding/duct damage in DPA and the maximum cladding temperature have been added. The thermal hydraulic module in place is the simple 1-D *thermo* module. Ideal orificing is used in this case, but the application of orificed flow is straightforward. Material damage is to be minimized to delay the point at which the cladding strain limit is reached, and the preference for the cladding temperature depends on the design goals: it should be maximized for thermal efficiency but minimized to reduce clad damage by FCCI.

The ACE regression performs very well on all dependent variables, as can be seen on the plot of regression results against the measured values in Figure 5.12. The largest average error among all dependent variables in the predictive test (trained on 90% of the measured data points) is 0.6%, found in the sodium density reactivity

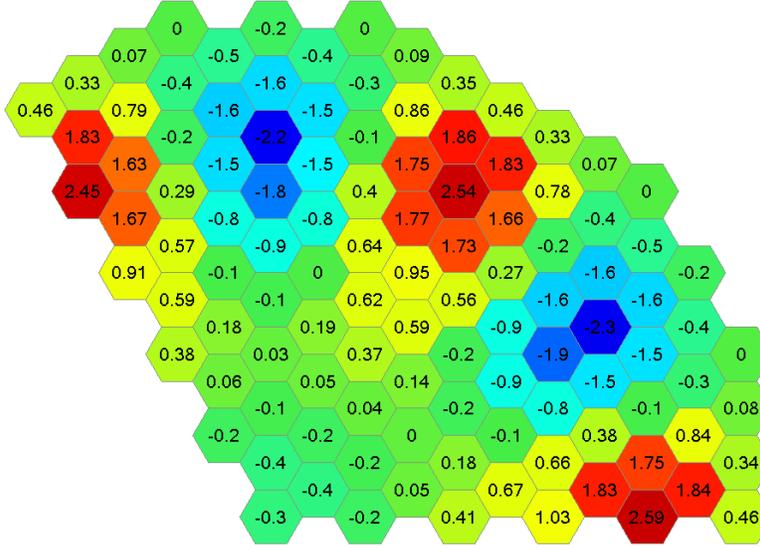


Figure 5.11: The 40th adjoint reference harmonic. The slight asymmetry in the  $1/6$  core is expected due to the asymmetric fuel management path. The localized nature of this high harmonic is evident. The adjoint harmonics are not normalized.

coefficient.

### 5.3.1 Comparison between direct and perturbation-based evaluations

To verify the validity of the ECPT-driven optimization engine, an identical test matrix was executed using direct EEC. The results between the two sets of runs are shown in Table 5.8. The optimal value for this set of preferences is on the upper bound of both the cladding thickness and the smear density design parameters. The optimal charge enrichment considering all these constraints is 2.72% by the direct case and 2.86% by the ECPT case. The effect of this 5% difference manifests through the design metrics as can be seen in the last column, resulting in an eventual 10% difference in the value of the aggregate objective function.

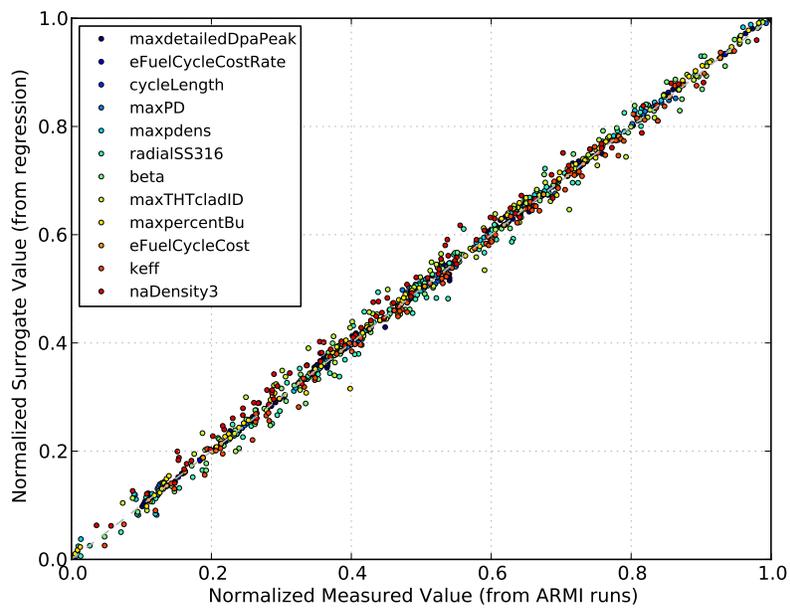


Figure 5.12: The ACE regression results for the high burnup fast reactor case.

Each direct calculation of the perturbed flux took DIF3D 25.2 seconds on average, while the MEPT calculations with DIFNT took 18.8 seconds. This speedup is far from as favorable as predicted by the operation counts in §3.1.6. It must be recalled, however, that DIFNT was built as a proof-of-concept code within hybrid C++/Python, whereas DIF3D is written as a production code in FORTRAN, and was highly optimized for speed by a national laboratory over decades. The Python interaction of DIFNT with ARMI is the speed bottleneck and can be sped up by optimizing the back-end treatments of macroscopic cross sections, mostly by porting it to C++. After the macroscopic cross sections are read, the computation of  $\delta L$ , the expansion coefficients, the perturbed flux, and the perturbed eigenvalue in a MEPT run takes 4.3 seconds. That DIF3D manipulates the macroscopic cross sections as well provides a proof of feasibility for dramatically reducing this bottleneck. The scalable and parallel nature of DIFNT makes it intuitive that its speed advantage for very large problems will be more pronounced.

Table 5.7: Physical Programming region boundaries for the high-burnup fast reactor optimization

Dependent Variable	Class	Highly Desirable	Desirable	Tolerable	Undesirable	Highly Undesirable
Sodium Density Coeff. ( $\zeta/^\circ C$ )	S	0.01	0.15	0.28	0.35	0.40
Max. Discharge Burnup (% FIMA)	S	25	28	31	35	40
Max. Power Density ( $W/cm^3$ )	L	250	130	120	110	100
Fuel Cycle Cost (\$ M/yr)	S	0.4	0.45	0.50	0.70	1.00
Max. Clad Damage (DPA)	S	250	450	600	650	700
Max. Clad Temperature ( $^\circ C$ )	L	510	505	500	490	480

162

Table 5.8: High-burnup reactor optimization results from direct and ECPT calculations. Results from the L-BFGS-B optimization engine are shown for both cases, although all others return nearly identical results.

Optimal parameter/metric	Direct	ECPT	Difference
Charge Enrichment. (wt. %)	2.724	2.863	5.12%
Cladding Thickness (mm)	0.599	0.599	0.00%
Smear Density (%)	69.99	69.99	0.00%
Sodium Density Coeff. ( $\zeta/^\circ C$ )	0.296	0.331	12.0%
Max. Discharge Burnup (% FIMA)	31.86	31.85	-0.03%
Max. Power Density ( $W/cm^3$ )	148.5	146.8	-1.10%
Fuel Cycle Cost (\$ M/yr)	0.7481	0.7759	3.73%
Max. Clad Damage (DPA)	611.1	608.9	-0.35%
Max. Clad Temperature ( $^\circ C$ )	509.5	509.5	0.01%
Objective Minimum	1.8950	2.082	9.90%

### 5.3.2 Demonstration of changing preferences

To demonstrate the flexibility and simplicity of the PP paradigm, we consider a change in the design preferences in this section. Where the preferences were set in Table 5.7 to maximize thermal efficiency and safety, we now consider a case where we prefer lower fuel cycle cost, lower cladding temperature, and have more leniency in the sodium reactivity coefficient, as specified in Table 5.9. Simply by adjusting the preferences accordingly, with no need to re-run the cases due to the existing ACE regression, a new optimization can be performed. The results shown in Table 5.10 show that the fuel cycle cost was reduced by 40%, the cladding temperature decreased, and the sodium density reactivity coefficient increased. Fuel cycle cost was reduced by decreasing the charge enrichment not to its lower limit of 0.3%, but rather to an intermediate value of 1.9%. A large reduction in the smear density also contributed to the reduced fuel cycle cost.

Overall, the design changes are substantial. While they agree with intuition, the manual determination of the precise optimal design values would require significant time and effort from the designer. The utility of the the entire system developed in this work can be clearly seen in this case, where a large change in design preferences occurred and the designer was able to rapidly and robustly determine the new optimal design considering many parameters and metrics.

Table 5.9: Alternate Physical Programming region boundaries for the high-burnup fast reactor optimization.

Dependent Variable	Class	Highly Desirable	Desirable	Tolerable	Undesirable	Highly Undesirable
Sodium Density Coeff. ( $^{\circ}C$ )	S	0.01	0.28	0.35	0.4	0.5
Max. Discharge Burnup (% FIMA)	S	25	31	36	38	40
Max. Power Density ( $W/cm^3$ )	L	250	130	120	110	100
Fuel Cycle Cost (\$ M/yr)	S	0.10	0.20	0.30	0.45	0.50
Max. Clad Damage (DPA)	S	250	450	600	650	700
Max. Clad Temperature ( $^{\circ}C$ )	S	480	490	500	505	510

Table 5.10: High-burnup reactor optimization results from the alternate preference set.

Optimal parameter/metric	ECPT results (alternate preferences)
Charge Enrichment (wt. %)	1.901
Cladding Thickness (mm)	0.2010
Smear Density (%)	61.91
Sodium Density Coeff. ( $\text{c}/^{\circ}C$ )	0.3743
Max. Discharge Burnup (% FIMA)	34.08
Max. Power Density ( $W/cm^3$ )	129.2
Fuel Cycle Cost (\$ M/yr)	0.4512
Max. Cladding Damage (DPA)	660.6
Max. Clad Temperature ( $^{\circ}C$ )	506.5
Objective Minimum	2.662

## CHAPTER VI

### Summary and Conclusions

A set of methods that allows designers to efficiently optimize high-burnup nuclear reactors has been developed. Because reactor design is a highly multi-objective practice, the methods are built within a new computational analysis framework that enables the automated calculation of a diverse set of design variables of interest. The net result is a tool that accepts a certain design space and a set of performance preferences and produces the optimal design.

A fast equilibrium method is required to evaluate the objective functions during optimization. The REBUS equilibrium method was shown to have significant inadequacies for the high-burnup reactors of interest, and was subsequently revisited and implemented as the Enhanced Equilibrium Cycle (EEC) method containing essential new features including fuel performance coupling, burnup dependent cross sections, and computational parallelism. The consideration of fuel performance proved to be a very important development in equilibrium cycles, with sensitivities on the order of 1000 pcm on the equilibrium  $k_{eff}$ . Additionally, an instability was discovered for

very high burnup cases.

To accelerate the equilibrium evaluations, a computationally-efficient mapping between fresh-core design parameters and equilibrium performance was developed, making use of a time-independent Modal Expansion Perturbation Theory (MEPT) coupled to EEC. The MEPT method builds an explicit first-order expression for the perturbed flux given any arbitrary perturbation in the steady-state neutron balance statement. The flux perturbation is expressed as an expansion of  $\lambda$ -eigenmodes, where the expansion coefficients are determined using biorthogonality of the reference real and adjoint flux harmonics. In order to resolve spatially-localized perturbations, MEPT requires large numbers of  $\lambda$ -eigenmodes to be calculated for the reference case, a feat accomplished with a Block Krylov Schur iterative eigenvalue solver in place of the traditional power method. The combination of MEPT and EEC, along with a first-order depletion approximation formed Equilibrium Cycle Perturbation Theory (ECPT), with all of the direct flux and most of the direct depletion calculations replaced with high-speed perturbation methods. ECPT allows rapid evaluations of performance metrics at an equilibrium state given a set of BOL design perturbations. The methods were implemented under the 3-D multigroup diffusion approximation in triangular geometry in an all new code called DIFNT.

A multidisciplinary design optimization framework was developed around the ECPT tool. In multi-objective optimization, the final result is dependent on the user's preferences. Weighting various objectives is traditionally an iterative process, requiring many guesses of weighting parameters before the results truly reflect the user's intentions. Physical Programming (PP) is a known approach that avoids

this iterative process, being designed to allow a single specification preferences in physically-meaningful terms, and was implemented here for reactor optimization. PP transforms the often-frustrating practice of MDO into a straightforward, step-by-step process that can be operated by non-specialist users. Combined with the generality and automation of ARMI, the full system allows each member of a design team to independently perform optimizations on design changes of interest with minimal training in optimization practice, therefore multiplying the productivity of the team.

Since most linear programming tools require continuous functions to optimize, the PP lexicon was implemented on a multivariate regression of a set of ECPT results. We chose the Alternating Conditional Expectation (ACE) algorithm to produce a surrogate model because it makes no assumptions of the functional forms of the behavior of the dependent variables. Once the surrogate was trained by multiple ECPT results, it was transformed into PP class functions, which were in turn summed into an aggregate objective function which was optimized using standard methods. The full system was used to optimize a Traveling Wave Reactor considering many performance metrics at once.

## 6.1 Potential Improvements

We showed that MEPT is capable of approximating large perturbations and that ECPT is capable of driving an optimization system, but there is room for improvement on the methods.

MEPT was implemented using the finite-difference formulation for simplicity,

but the nodal formulation of neutron transport is the industry standard for core calculations. Though the nodal formulation is completely different from the finite difference formulation, the resulting matrix equations are of the same form, and the application of the Krylov Schur solver may remain unchanged. The efficacy of MEPT on coarse-mesh nodal cases compared to finite difference cases is not obvious, and requires study.

We showed that if the perturbed eigenvalue is known, the maximum point flux error may be reduced by more than 50% from the assumption that it remains constant. Since the perturbed  $M$  and  $B$  matrices must be computed to determine the expansion coefficients, which in turn determine the perturbed flux, the perturbed eigenvalue may be estimated before the perturbed flux is determined. Thus, a higher-order accuracy may be achieved without incurring a substantial increase in computation.

In the current implementation, building the perturbed macroscopic cross sections from the microscopic library and the perturbed number densities is the computational bottleneck during a ECPT run. The data management system, ARMI, performs this simple operation, but involves overhead to enable thermal hydraulic coupling and pin-level geometry details. This overhead slows down such a multi-level summation, but the issue can be readily solved by moving more of the code from Python into a compiled language such as C++. The same generation of macroscopic cross sections in DIF3D (coded in FORTRAN), for example, is nearly instantaneous.

While developed with the parallel matrix library Trilinos, reference harmonics for DIFNT have not been attempted in parallel. Spreading the inner and outer iterations involved in a MEPT reference calculation over many processors and chassis in a

compute cluster would allow very large problems to be solved with ECPT. Fine-mesh problems will likely require more harmonics than did the cases studied in this work to perform well, and parallel code will be necessary to determine any thresholds of applicability. Most design optimization can be done with the coarse meshes studied, but enabling advanced applications and fine-tuning on very high-resolution models requires study. Associated with this work will be the building of DIFNT for 64-bit machines, which has been accomplished on a Linux platform but not on Windows.

Because of the explicit nature of MEPT, any parameter available to ARMI is a possible dependent variable for optimization. Only a few such metrics were considered in this work. While sufficient for demonstrating the methods, more meaningful optimization requires the consideration of more metrics. The ability of ARMI to automatically generate key parameters during full transients through the SAS4A/SASSYS code for arbitrary perturbations is particularly useful for optimizing against safety. Other highly interesting parameters include thermal striping under orificed flow conditions, fuel burnup limits, duct distortion, and cladding wastage. Such detailed optimizations will be undergone in the near future. Relatedly, optimizing other independent variables than those shown could provide better and broader optimization. Any possible design change is captured by  $\delta L$  and therefore handled by ECPT. The ACE algorithm is capable of handling discontinuous jump changes in specific independent variables. Design parameters involved with the fuel management during an equilibrium cycle could be treated in the regression with this capability, allowing the optimization of the equilibrium fuel paths. There are no technical barriers within ECPT precluding its use to optimize fuel management as

well as lattice and fuel designs.

## 6.2 Alternative Applications

Besides optimizing reactors, the methods developed have other potential applications. EEC can naturally be used for all the purposes that traditional equilibrium methods are used for. To be useful for closed-cycle analysis, an external cycle module will need to be added to ARMI, allowing it to also make use of the fuel performance information and parallelism. Being ex-core, the closed-cycle aspects of the code will not be affected by the burnup-dependent microscopic cross section treatment.

Time-independent MEPT is applicable to many of the traditional uses of perturbation theory. Additionally, its ability to explicitly form the perturbed flux could enable new applications in transient codes coupled to accelerated, perturbation-based spatial kinetics. Rather than performing full neutronics solutions at each time step during a coupled transient, a system could detect the deviation from reference and perform updated reference calculations only when a certain threshold is violated. Such a system could conceivably speed up a transient code substantially. Another application of MEPT could be found in online core-following or in training systems such as core simulators. Detailed flux updates could be performed in nearly real-time to give operators or trainees insight into the complex happenings during transients or the nominal core lifetime.

ECPT is conducive to uncertainty quantification studies. The nuclear data community has rigorous methods to determine uncertainties due to data at a point in time [107], but the ability to propagate these uncertainties through time is still a re-

search topic. Besides only focusing on data uncertainties, other uncertainties, such as those within hot channel factors, manufacturing tolerances, model approximations, etc. could also be propagated to equilibrium with ECPT.

## REFERENCES

## REFERENCES

- [1] A. M. WEINBERG, *The First Nuclear Era: The Life and Times of a Technological Fixer*, AIP Press (1994).
- [2] I.A.E.A., “Nuclear Technology Review,” IAEA/NTR/2011, International Atomic Energy Agency (2011).
- [3] H. P. PLANCHON, J. I. SACKETT, G. H. GOLDEN, and R. H. SEVY, “Implications of the EBR-II inherent safety demonstration test,” *Nuclear Engineering and Design*, **101**, 75 (1987).
- [4] T. L. SCHULZ, “Westinghouse AP1000 advanced passive plant,” *Nuclear Engineering and Design*, **236**, 1547 (2006).
- [5] Nuclear Energy Agency, *Uranium 2009: Resources, Production and Demand*, OECD (2010).
- [6] British Petroleum, “Statistical Review of World Energy,” <http://www.bp.com/statisticalreview>, 2011.
- [7] S. M. FEINBERG, “Discussion Comment,” *Rec. of Proc. Session B-10, ICP UAE, United Nations, Geneva, Switzerland* (1958).
- [8] B. ATEFI, M. J. DRISCOLL, and D. D. LANNING, *Evaluation of the breed/burn fast reactor concept*, PhD thesis, Massachusetts Inst. of Tech., 1979.
- [9] G. J. FISCHER, A. GALPERIN, S. SHENOY, and B. ATEFI, “Fast-mixed spectrum reactor. Progress report for 1980,” Brookhaven National Lab. (1980).
- [10] E. TELLER, M. ISHIKAWA, L. WOOD, R. HYDE, and J. NUCKOLLS, “Completely automated nuclear reactors for long-term operation II: Toward a conceptual-level point design of a high-temperature, gas-cooled central power station system,” Lawrence Livermore National Lab., CA (1996).

- [11] H. SEKIMOTO, K. RYU, and Y. YOSHIMURA, “CANDLE: The new burnup strategy,” *Nucl. Sci. Eng.*, **139**, 306 (2001).
- [12] R. C. PETROSKI, *General Analysis of Breed-and-Burn Reactors and Limited-Separations Fuel Cycles*, PhD thesis, Massachusetts Inst. of Tech., 2010.
- [13] T. ELLIS et al., “Traveling-wave reactors: a truly sustainable and full-scale resource for global energy needs,” *Proceedings of ICAPP*, **10**, 13 (2010).
- [14] G. L. HOFMAN, L. C. WALTERS, and T. H. BAUER, “Metallic fast reactor fuels,” *Progress in Nuclear Energy*, **31**, 83 (1997).
- [15] D. S. GELLES, “Microstructural examination of commercial ferritic alloys at 200 DPA,” *Journal of Nuclear Materials*, **233**, 293 (1996).
- [16] L. C. WALTERS, “Thirty years of fuels and materials information from EBR-II,” *Journal of Nuclear Materials*, **270**, 39 (1999).
- [17] A. M. WEINBERG and F. P. WIGNER, *The physical theory of neutron chain reactions*, Univ. of Chicago Press, Chicago (1958).
- [18] P. ROUSSOPOLOS, “Méthodes variationnelles en théorie des collisions,” *C. R. Acad. Sci. Paris*, **236**, 1858 (1953).
- [19] G. C. POMRANING, “A variational principle for linear systems,” *Journal of the Society for Industrial and Applied Mathematics*, **13**, 511 (1965).
- [20] G. C. POMRANING, “Variational Principle for Eigenvalue Equations,” *Journal of Mathematical Physics*, **8**, 149 (1967).
- [21] L. N. USACHEV, “Perturbation theory for the breeding ratio and for other number ratios pertaining to various reactor processes,” *Journal of Nuclear Energy*, **18**, 571 (1964).
- [22] W. M. STACEY JR, “Variational estimates and generalized perturbation theory for the ratios of linear and bilinear functionals,” *Journal of Mathematical Physics*, **13**, 1119 (1972).
- [23] W. M. STACEY JR, “Variational estimates of reactivity worths and reaction rate ratios in critical nuclear reactors,” *Nucl. Sci. Eng.*, **48**, 444 (1972).
- [24] A. GANDINI, “Implicit and explicit higher order perturbation methods for nuclear reactor analysis,” *Nucl. Sci. Eng.*, **67**, 347 (1978).

- [25] G. PALMIOTTI, “Use of the explicit high-order perturbation formulation,” *Nucl. Sci. Eng.*, **83**, 281 (1983).
- [26] H. MITANI, “Higher order perturbation method in reactor calculations,” *Nucl. Sci. Eng.*, **51**, 180 (1973).
- [27] R. AIGLE, G. B. BRUNA, and A. SARGENI, “On the completeness of the multigroup eigenfunctions set of a reactor system Boltzmann operator,” *Annals of Nuclear Energy*, **21**, 445 (1994).
- [28] G. PALMIOTTI, “The FRENCH (flux reconstructed explicitly by neutronic calculated harmonics) method in multidimensional and multigroup theory,” *Annals of Nuclear Energy*, **14**, 167 (1987).
- [29] J. MOREIRA and J. C. LEE, “Space-time analysis of reactor control rod worth measurements,” *Nucl. Sci. Eng.*, **86**, 91 (1984).
- [30] J. MOREIRA and J. C. LEE, “Accuracy of the modal-local method for reactivity determination,” *Nucl. Sci. Eng.*, **98**, 244 (1988).
- [31] A. GANDINI, “Time-dependent Generalized Perturbation Methods for Burn-up Analysis,” *CNEN RT/FI (75)*, **4** (1975).
- [32] J. M. KALLFELZ, G. BRUNA, G. PALMIOTTI, and M. SALVATORES, “Burn-up Calculations with Time-Dependent Generalized Perturbation Theory,” *Nucl. Sci. Eng.*, **62**, 304 (1977).
- [33] A. GANDINI, M. SALVATORES, and L. TONDINELLI, “New developments in generalized perturbation methods in the nuclide field,” *Nucl. Sci. Eng.*, **62**, 339 (1977).
- [34] M. L. WILLIAMS and C. R. WEISBIN, “Sensitivity and Uncertainty Analysis for Functionals of the Time-Dependent Nuclide Density Field,” ORNL-5393, Oak Ridge National Lab. (1978).
- [35] M. L. WILLIAMS, “Development of depletion perturbation theory for coupled neutron/nuclide fields,” ORNL/TM-5958, Oak Ridge National Lab. (1978).
- [36] J. R. WHITE, “Application of depletion perturbation theory to fuel cycle burnup analysis,” *Trans. Am. Nucl. Soc.*, **12** (1979).
- [37] W. S. YANG and T. J. DOWNAR, “Generalized Perturbation Theory for Constant Power Core Depletion,” *Nucl. Sci. and Eng.*, **99**, 353 (1988).

- [38] H. B. CHOI and T. J. DOWNAR, “Sensitivity theory for the closed nuclear fuel cycles,” *Nucl. Sci. Eng.*, **111**, 205 (1992).
- [39] R. VAN GEEMERT and J. E. HOOGENBOOM, “Development of parallelized higher-order generalized depletion perturbation theory for application in equilibrium cycle optimization,” *Annals of Nuclear Energy*, **28**, 1377 (2001).
- [40] A. GANDINI, M. SALVATORES, and G. SENA, “Use of generalized perturbation methods for optimization of reactor design,” *Journal of Nuclear Energy*, **23**, 469 (1969).
- [41] C. P. TZANOS, E. P. GYFTOPOULOS, and M. J. DRISCOLL, “Optimization of material distributions in fast reactor cores,” *Nucl. Sci. Eng.*, **52**, 84 (1973).
- [42] W. B. TERNEY and E. A. WILLIAMSON JR, “The design of reload cores using optimal control theory,” *Nucl. Sci. Eng.*, **82**, 260 (1982).
- [43] C. R. DRUMM and J. C. LEE, “Gadolinium burnable absorber optimization by the method of conjugate gradients,” *Nucl. Sci. Eng.*, **96**, 17 (1987).
- [44] W. LI-MING, *Optimal burnable poison loading strategy*, PhD thesis, University of Michigan, 1990.
- [45] D. H. JACOBSON, M. M. LELE, and J. L. SPEYER, “New necessary conditions of optimality for control problems with state-variable inequality constraints,” *Journal of Mathematical Analysis and Applications*, **35**, 255 (1971).
- [46] R. T. SORENSEN, *Systematic method for optimizing plutonium transmutation in LWRs*, PhD thesis, University of Michigan, 2007.
- [47] J. C. DAVIS, *Optimizing SFR Transmutation Performance Through Direct Adjoining Control Theory*, PhD thesis, University of Michigan, 2007.
- [48] M. D. DECHAINED and M. A. FELTUS, “Nuclear fuel management optimization using genetic algorithms,” *Nuclear Technology*, **111**, 109 (1995).
- [49] D. J. KROPACZEK and P. J. TURINSKY, “In-core nuclear fuel management optimization for pressurized water reactors utilizing simulated annealing,” *Nuclear Technology*, **95**, 9 (1991).

- [50] J. L. C. CHAPOT, F. CARVALHO DA SILVA, and R. SCHIRRU, “A new approach to the use of genetic algorithms to solve the pressurized water reactor’s fuel management optimization problem,” *Annals of Nuclear Energy*, **26**, 641 (1999).
- [51] J. KREPEL, S. PELLONI, K. MIKITYUK, and P. CODDINGTON, “EQL3D: ERANOS based equilibrium fuel cycle procedure for fast reactors,” *Annals of Nuclear Energy*, **36**, 550 (2009).
- [52] A. P. OLSON, “A users guide for the REBUS-PC code, version 1.4,” ANL/RERTR/TM-32, Argonne National Lab. (2002).
- [53] N. W. TOURAN, J. C. CHEATHAM, and R. C. PETROSKI, “Model Biases in High-Burnup Fast Reactor Simulations,” *Proceedings of PHYSOR* (2012).
- [54] J. S. WARSA, T. A. WAREING, J. E. MOREL, J. M. MCGHEE, and R. B. LEHOUCQ, “Krylov subspace iterations for deterministic k-eigenvalue calculations,” *Nucl. Sci. Eng.*, **147**, 26 (2004).
- [55] L. BREIMAN and J. H. FRIEDMAN, “Estimating optimal transformations for multiple regression and correlation,” *Journal of the American Statistical Association*, **80**, 580 (1985).
- [56] A. MESSAC, “Physical programming: effective optimization for computational design,” *AIAA journal*, **34**, 149 (1996).
- [57] R. P. HOSTENY, “ARC System fuel cycle analysis capability, REBUS-2,” ANL-7721, Argonne National Lab. (1978).
- [58] W. B. WILSON, S. T. COWELL, T. R. ENGLAND, A. C. HAYES, and P. MOLLER, “A Manual for CINDER’90 Version 07.4 Codes and Data,” LA-UR-99-361, Los Alamos National Lab. (2008).
- [59] K. L. DERSTINE, “DIF3D: a code to solve one-, two-, and three-dimensional finite-difference diffusion theory problems,” ANL-82-64, Argonne National Lab. (1984).
- [60] B. J. TOPPEL, A. L. RAGO, and D. M. O’SHEA, “MC2: A Code to Calculate Multigroup Cross Sections,” ANL-7318 (1967).
- [61] B. M. CARMICHAEL, “Standard interface files and procedures for reactor physics codes, version III,” LA-5486-MS, Los Alamos National Lab. (1974).

- [62] J. HENDRICKS, “MCNPX version 2.5. c,” LA-UR-03-2202, Los Alamos National Lab. (2003).
- [63] R. C. PETROSKI, “Design of a 2400MW Liquid-Salt Cooled Flexible Conversion Ratio Reactor,” Master’s thesis, Massachusetts Inst. of Tech., 2008.
- [64] N. W. TOURAN et al., “Technical Considerations and Capabilities of a Near-Term Deployable Traveling Wave Reactor Core,” *Trans. Am. Nucl. Soc.*, **104**, 934 (2011).
- [65] K. L. BASEHORE and N. E. TODREAS, “SUPERENERGY-2: a multiassembly, steady-state computer code for LMFBR core thermal-hydraulic analysis,” PNL-3379, Battelle Pacific Northwest Labs. (1980).
- [66] J. W. JACKSON and N. E. TODREAS, “COBRA IIIc/MIT-2: a digital computer program for steady state and transient thermal-hydraulic analysis of rod bundle nuclear fuel elements. Final report,” Massachusetts Inst. of Tech. (1981).
- [67] J. E. CAHALAN, A. M. TENTNER, and E. E. MORRIS, “Advanced LMR safety analysis capabilities in the SASSYS-1 and SAS4A computer codes,” ANL/RA/CP-79955, Argonne National Lab. (1994).
- [68] M. B. CHADWICK et al., “ENDF/B-VII. 1 Nuclear Data for Science and Technology: Cross Sections, Covariances, Fission Product Yields and Decay Data,” *Nuclear Data Sheets*, **112**, 2887 (2011).
- [69] A. I. KARAHAN and J. BUONGIORNO, “A new code for predicting the thermo-mechanical and irradiation behavior of metallic fuels in sodium fast reactors,” *Journal of Nuclear Materials*, **396**, 283 (2010).
- [70] G. RIMPAULT, “Algorithmic features of the ECCO cell code for treating heterogeneous fast reactor assemblies,” in *Proc. Intl Conf on Mathematics and Computations, Reactor Physics, and Environmental Analyses, Portland, Oregon, USA*, pgs. 1–5, 1995.
- [71] A. P. OLSON, “Status of Rebus fuel management software development for RERTR applicaitons,” ANL-RERTR-TM-32, Argonne National Lab., Argonne, IL (2000).
- [72] G. I. BELL and S. GLASSTONE, *Nuclear reactor theory*, Van Nostrand Reinhold Company (1979).

- [73] W. STACEY and I. NETLIBRARY, *Nuclear reactor physics*, volume 2, Wiley Online Library (2007).
- [74] R. S. MODAK, H. P. GUPTA, and V. K. JAIN, “A scheme for the evaluation of  $\lambda$ -modes of a neutron diffusion equation,” *Annals of Nuclear Energy*, **21**, 195 (1994).
- [75] R. S. MODAK and A. GUPTA, “New applications of Orthomin (1) algorithm for k-eigenvalue problem in Reactor Physics,” *Annals of Nuclear Energy*, **33**, 538 (2006).
- [76] J. J. E. SO, T. J. DOWNAR, R. JANARDHAN, and H. J. SIEGEL, “Mapping conjugate gradient algorithms for neutron diffusion applications onto SIMD, MIMD, and mixed-mode machines,” *International journal of parallel programming*, **26**, 183 (1998).
- [77] R. MODAK and S. DEGWEKER, “Convergence of the iteration scheme in the nodal expansion method for the solution of the diffusion equation,” *Annals of Nuclear Energy*, **13**, 539 (1986).
- [78] S. MENON and D. KHANDEKAR, “Convergence of the inner-outer iteration scheme,” *Zeitschrift für Angewandte Mathematik und Physik*, **35**, 321 (1984).
- [79] Z. I. WONICKI, “The numerical analysis of eigenvalue problem solutions in the multigroup neutron diffusion theory,” *Progress in Nuclear Energy*, **33**, 301 (1998).
- [80] Y. CAO, *Space-Time Kinetics and Time-Eigenfunctions.*, PhD thesis, Univ. of Michigan, 2008.
- [81] L. N. TREFETHEN and D. BAU, *Numerical linear algebra*, Society for Industrial and Applied Mathematics (1997).
- [82] D. C. SORENSEN, “Implicit application of polynomial filters in a k-step Arnoldi method,” *SIAM J. Matrix Anal. Appl.*, **13**, 357 (1992).
- [83] R. B. LEHOUCQ, D. C. SORENSEN, and C. YANG, *ARPACK users’ guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, volume 6, SIAM (1998).
- [84] M. HEROUX et al., “An Overview of Trilinos,” SAND2003-2927, Sandia National Lab. (2003).

- [85] G. W. STEWART, “A Krylov–Schur Algorithm for Large Eigenproblems,” *SIAM Journal on Matrix Analysis and Applications*, **23**, 601 (2001).
- [86] G. VERDÚ, R. MIRÓ, D. GINESTAR, and V. VIDAL, “The implicit restarted Arnoldi method, an efficient alternative to solve the neutron diffusion equation,” *Annals of nuclear energy*, **26**, 579 (1999).
- [87] M. LUTZ, *Programming Python*, O’Reilly Media, Inc. (2010).
- [88] M. A. HEROUX and J. M. WILLENBRING, “Trilinos Users Guide,” SAND2003-2952, Sandia National Lab. (2003).
- [89] M. A. HEROUX, “AztecOO User Guide,” SAND2004-3796, Sandia National Lab. (2007).
- [90] M. SALA, “Amesos 2.0 reference guide,” SAND-4820, Sandia National Lab. (2004).
- [91] G. G. DAVIDSON, T. M. EVANS, J. J. JARRELL, and R. N. SLAYBAUGH, “Massively Parallel, Three-Dimensional Transport Solutions for the k-Eigenvalue Problem,” *Proceedings of the ANS Topical Conference in Mathematics and Computation* (2011).
- [92] E. K. LEE, Y. H. KIM, K. H. CHA, and M. G. PARK, “Reconstruction of core axial power shapes using the alternating conditional expectation algorithm,” *Annals of Nuclear Energy*, **26**, 983 (1999).
- [93] A. VARUTTAMASENI, *Bayesian network representing system dynamics in risk analysis of nuclear systems*, PhD thesis, University of Michigan, 2011.
- [94] I. E. FRANK and S. LANTERI, “ACE: A non-linear regression model,” *Chemometrics and Intelligent Laboratory Systems*, **3**, 301 (1988).
- [95] J. H. FRIEDMAN and W. STUETZLE, “Smoothing of scatterplots,” ORION-003, Standfort University (1982).
- [96] D. WANG and M. MURPHY, “Estimating optimal transformations for multiple regression using the ACE algorithm,” *Journal of Data Science*, **2**, 329 (2004).
- [97] P. SPECTOR, J. H. FRIEDMAN, R. TIBSHIRANI, and T. LUMLEY, “Package acepack,” 2012.

- [98] K. HORNIK, “The R FAQ,” 2011, <http://CRAN.R-project.org/doc/FAQ/R-FAQ.html>.
- [99] R. MARLER and J. ARORA, “Survey of multi-objective optimization methods for engineering,” *Structural and multidisciplinary optimization*, **26**, 369 (2004).
- [100] A. MESSAC and A. ISMAIL-YAHAYA, “Multiobjective robust design using physical programming,” *Structural and Multidisciplinary Optimization*, **23**, 357 (2002).
- [101] S. G. NASH, “Newton-type minimization via the Lanczos method,” *SIAM Journal on Numerical Analysis*, **21**, 770 (1984).
- [102] R. H. BYRD, P. LU, J. NOCEDAL, and C. ZHU, “A limited memory algorithm for bound constrained optimization,” *SIAM Journal on Scientific Computing*, **16**, 1190 (1995).
- [103] C. ZHU, R. H. BYRD, P. LU, and J. NOCEDAL, “Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization,” *ACM Transactions on Mathematical Software (TOMS)*, **23**, 550 (1997).
- [104] M. J. D. POWELL, “A direct search optimization method that models the objective and constraint functions by linear interpolation,” *Advances in optimization and numerical analysis*, **7**, 51 (1994).
- [105] D. KRAFT, “Algorithm 733: TOMP–Fortran modules for optimal control calculations,” *ACM Transactions on Mathematical Software (TOMS)*, **20**, 262 (1994).
- [106] S. KIRKPATRICK, C. D. GELATT JR, and M. P. VECCHI, “Optimization by simulated annealing,” *Science*, **220**, 671 (1983).
- [107] G. ALIBERTI et al., “Nuclear data sensitivity, uncertainty and target accuracy assessment for future nuclear systems,” *Annals of Nuclear Energy*, **33**, 700 (2006).