

ADJOINT-BASED UNCERTAINTY QUANTIFICATION AND SENSITIVITY
ANALYSIS FOR REACTOR DEPLETION CALCULATIONS

A Dissertation
by
HAYES FRANKLIN STRIPLING

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Marvin L. Adams
Committee Members,	Bani K. Mallick
	Ryan G. McClaren
	Jim E. Morel
	Mihai Anitescu
Department Head,	Yassin Hassan

August 2013

Major Subject: Nuclear Engineering

Copyright 2013 Hayes Franklin Stripling

ABSTRACT

Depletion calculations for nuclear reactors model the dynamic coupling between the material composition and neutron flux and help predict reactor performance and safety characteristics. In order to be trusted as reliable predictive tools and inputs to licensing and operational decisions, the simulations must include an accurate and holistic quantification of errors and uncertainties in its outputs. Uncertainty quantification is a formidable challenge in large, realistic reactor models because of the large number of unknowns and myriad sources of uncertainty and error.

We present a framework for performing efficient uncertainty quantification in depletion problems using an adjoint approach, with emphasis on high-fidelity calculations using advanced massively parallel computing architectures. This approach calls for a solution to two systems of equations: (a) the forward, engineering system that models the reactor, and (b) the adjoint system, which is mathematically related to but different from the forward system. We use the solutions of these systems to produce sensitivity and error estimates at a cost that does not grow rapidly with the number of uncertain inputs. We present the framework in a general fashion and apply it to both the source-driven and k-eigenvalue forms of the depletion equations. We describe the implementation and verification of solvers for the forward and adjoint equations in the PDT code, and we test the algorithms on realistic reactor analysis problems. We demonstrate a new approach for reducing the memory and I/O demands on the host machine, which can be overwhelming for typical adjoint algorithms. Our conclusion is that adjoint depletion calculations using full transport solutions are not only computationally tractable, they are the most attractive option for performing uncertainty quantification on high-fidelity reactor analysis problems.

DEDICATION

I dedicate this dissertation to my loving and beautiful wife, Jenny Stripling, who believes in and supports me without exception. Also to my mother and father, Tammy and Hayes, and brother, Ross, who always encourage me to pursue goals that lead to happiness and provide the means to get there.

ACKNOWLEDGEMENTS

First, I wish to express my warm appreciation to Dr. Marvin Adams, my friend and advisor, for his dedication to this research and to my personal and professional development over the last 5 years. His mentorship led me through doors and down paths that simply were not available before we began working together, and I am forever grateful. I am also thankful for Mihai Anitescu for his technical guidance, which led directly to the content in this dissertation, and for the time we spent collaborating at Argonne National Laboratory. I wish to acknowledge Dr. Ryan McClaren, who made great efforts to provide career advice and opportunities, some of which were the most memorable and exciting experiences as a graduate student. I am grateful for advice, insight, and teachings of Drs. Jim Morel and Bani Mallick and for their contributions to this dissertation while on my committee. Finally, I greatly appreciate Dr. Kord Smith, Massachusetts Institute of Technology, for his guidance as a technical leader on this project and for the time he spent editing this dissertation.

I am thankful for the nuclear engineering community and department at Texas A&M University. I benefited greatly from the tireless work of the faculty and staff, who shape the department with each student's education as priority one. A big, special thanks for Mr. W. Daryl Hawkins for his constant support and patience during the implementation phase of this research. I have especially enjoyed working and becoming friends with several fellow students, many of whom will be lifelong friends and colleagues.

I extend a special acknowledgment to the Department of Energy's Computational Science Graduate Fellowship program (DOE CSGF - grant number DE-FG02-

97ER25308). This program provides outstanding support, resources, and networking opportunities to doctoral students pursuing research in computational science and high-performance computing. Their emphasis on professional development and a broadened program of study created new and unexpected opportunities during my time as a graduate student. I also wish to acknowledge funding and support from the Center for Exascale Simulation for Advanced Reactors (CESAR), contract no. DE-AC02-06CH11357, which is focused on the co-design of next-generation computer software and architectures for reactor analysis calculations.

NOMENCLATURE

ANL	Argonne National Laboratory
BOEC	Beginning of equilibrium cycle
CDS	Convergent-divergent shuffling
DAE	Differential-algebraic equation
DFEM	Discontinuous finite-element method (spatial discretization)
DG	Discontinuous Galerkin (spatial discretization)
EOEC	End of equilibrium cycle
FIMA	Fissions per initial metal atom
FLOP	Floating point operation
GPT	Generalized perturbation theory
I/O	Input/Output
ICS	Inward-convergent shuffling
MOEC	Middle of equilibrium cycle
MSA	Method of successive approximations
NSC	Nuclear science center
ODE	Ordinary Differential Equation
OS	Operator Splitting
PDT	Parallel or Particle Deterministic Transport Code
PS&E	Predictive science and engineering
QOI	Quantity of interest
RK	Runge-Kutta (time discretization)
UQ	Uncertainty quantification
UQSA	Uncertainty quantification and sensitivity analysis

TABLE OF CONTENTS

	Page
ABSTRACT	ii
DEDICATION	iii
ACKNOWLEDGEMENTS	iv
NOMENCLATURE	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	xi
LIST OF TABLES	xv
1. INTRODUCTION	1
1.1 The Challenge of Uncertainty Quantification in Predictive Science . .	2
1.2 The Adjoint Approach for Generating Uncertainty and Error Information	9
1.3 Main Contributions of this Research	13
1.4 A Preview of the Remaining Sections	14
2. PRELIMINARIES	16
2.1 The Depletion Equations	17
2.2 Differential-Algebraic Equations	22
2.3 Runge-Kutta Time Discretization Methods	24
2.4 The Adjoint Bateman and Neutron Transport Operators	32
2.4.1 An example problem to motivate the adjoint approach	32
2.4.2 The form of the adjoint operators corresponding to the forward transport equation	37
2.4.3 Modification of the forward solver to solve for the adjoint flux	40
2.4.4 The adjoint Bateman operator	44
2.5 Adjoint Consistency of Spatial and Temporal Discretizations	47

2.5.1	Discontinuous Galerkin spatial discretization	47
2.5.2	Runge-Kutta time discretization	52
2.6	Review of the Literature Leading to the Current State of the Art in Depletion Perturbation Theory	61
3.	THEORY	64
3.1	A Variational Derivation of the Adjoint Equations Corresponding to Parameter-Dependent DAE Systems	65
3.1.1	Adjoint-based QOI sensitivity estimates	67
3.1.2	Adjoint-based QOI error estimates	74
3.1.3	A discussion of the adjoint formalism	83
3.2	The Application of our Framework to the Source-Driven Depletion Equations	87
3.3	The Application of our Framework to the k-Eigenvalue Depletion Equations	94
3.4	Example of Extensibility: Addition of Heat Transfer Physics	105
3.5	Technique for Integrating the Sensitivity Equation	111
3.6	Adjoint Equations for Three Particular QOIs and Non-Conforming QOIs	113
3.6.1	The terminal inventory QOI	113
3.6.2	The terminal reaction rate QOI	114
3.6.3	The terminal reactivity QOI	117
3.6.4	Non-conforming QOIs	118
4.	STRATEGIES AND IMPLEMENTATION FOR MASSIVELY PARALLEL ARCHITECTURES	120
4.1	Description of the PDT Software: A Massively Parallel Discrete Ordinates Transport Solver	121
4.2	Description of the Depletion Solver in PDT	126
4.2.1	The operator splitting solution technique	126
4.2.2	The Runge-Kutta solution technique	129
4.3	Description of the Adjoint Transport Solver in PDT	131
4.4	Description of the Method of Successive Approximations Implementation in PDT	134
4.5	Description of the Depletion-Perturbation Solver in PDT	144
4.6	Schemes for Checkpointing the Forward Solution	149
4.6.1	Motivation for efficient checkpointing: An example problem .	149
4.6.2	Checkpointing source moments as a representation of the angular flux	150
4.6.3	Computational analysis of five checkpointing schemes	152

5. VERIFICATION AND TEST PROBLEMS	164
5.1 Verification Problems Targeting the Depletion Solver	166
5.1.1 A single component, infinite medium depletion verification problem in k-eigenvalue form	166
5.1.2 A 33 group model of the ^{239}Pu production chain in an infinite medium	170
5.2 Verification Problems Targeting the Steady-State Adjoint Transport Solver	177
5.2.1 Infinite medium problem with analytic sensitivities	177
5.2.2 Source-driven, 33g detector response problem	180
5.3 A Two-Group, Infinite Medium k-Eigenvalue Problem Targeting the Full Depletion Perturbation Solver	186
5.3.1 A walk-through of the solution procedure	186
5.3.2 Convergence rates of the sensitivity and error estimates	212
5.4 Two Source-Driven Test Problems for Verifying the Accuracy of the Parameter Derivatives	226
6. APPLICATION AND SCALING PROBLEMS	231
6.1 A One-Dimensional Traveling Wave Reactor Benchmark Problem with Depletion and Uncertainty Quantification	232
6.1.1 Problem description and reference results	232
6.1.2 Approach and solution using the PDT code	236
6.1.3 An equilibrium-cycle uncertainty quantification study	250
6.1.4 A multi-cycle uncertainty quantification study	256
6.2 Steady-State and Depletion Simulations of the Nuclear Science Center Research Reactor	260
6.2.1 NSC beginning-of-life sensitivity calculations	260
6.2.2 NSC depletion calculations	269
6.2.3 NSC depletion-perturbation calculations	280
6.3 Checkpointing Scheme Scaling Study	284
6.3.1 Description of the source-driven test problem	284
6.3.2 Scaling results	287
7. SUMMARY AND CONCLUSIONS	302
7.1 Summary	302
7.2 Conclusions	306
REFERENCES	310

APPENDIX A. A VARIATIONAL DERIVATION OF THE ADJOINT DEPLETION EQUATIONS CORRESPONDING TO THE SOURCE DRIVEN AND K-EIGENVALUE BURNUP EQUATIONS	317
A.1 The Source-Driven Case	317
A.2 The k-Eigenvalue Case	326
APPENDIX B. DETAILED DESCRIPTION OF THE DEPLETION, ADJOINT , AND DEPLETION PERTURBATION SOLVER IMPLEMENTATIONS IN PDT	340
B.1 User's Manual	341
B.2 Code Documentation	356
APPENDIX C. LISTING OF OTHER VERIFICATION PROBLEMS	384
C.1 Infinite Medium, Pure Absorber Source-Driven Problem	384
C.2 Infinite Medium, Source-Driven Problem with Scattering	392
APPENDIX D. THE FUNDAMENTAL SOLUTION MATRIX METHOD APPLIED TO FORWARD AND ADJOINT SYSTEMS	404
APPENDIX E. LISTING OF RUNGE-KUTTA METHODS	409
APPENDIX F. LISTING OF DEPLETION CHAINS	412
F.1 Synthetic Two Group, Four Component Model	412
F.2 33 group, fast spectrum cross sections from Argonne National Laboratory	413

LIST OF FIGURES

FIGURE	Page
4.1 Pseudocode for the operator splitting depletion algorithm	127
4.2 Pseudocode for the Runge-Kutta depletion algorithm	130
4.3 Pseudocode for the steady-state adjoint solver	133
4.4 Pseudocode for the MSA solver	144
4.5 Pseudocode for integrating the adjoint equations	145
4.6 Illustration of a general checkpointing scheme	148
4.7 Legend of symbols for checkpointing scheme cost analysis	153
4.8 Illustration of the STOR_ALL and STOR_MOM schemes.	154
4.9 Illustration of the CKPT_ALL and CKPT_MOM schemes.	157
4.10 Illustration of the INPT_MOM checkpointing scheme	159
5.1 Convergence rates of the Runge-Kutta schemes for infinite medium problem	170
5.2 Convergence rate of the explicit Euler scheme	174
5.3 Convergence rate of the modified Euler scheme	175
5.4 Convergence rate of the Runge-Kutta 4 scheme	176
5.5 Geometry of the 33 group steady-state adjoint verification problem. .	180
5.6 Convergence rates for $\frac{dQ}{d\sigma_t}$ in the 33 group steady-state adjoint transport problem	182
5.7 Convergence rates for $\frac{dQ}{d\sigma_s}$ in the 33 group steady-state adjoint transport problem	183
5.8 Convergence rates for $\frac{dQ}{d\sigma_f}$ in the 33 group steady-state adjoint transport problem	184
5.9 Convergence rates for $\frac{dQ}{d\nu}$ in the 33 group steady-state adjoint transport problem	185

5.10	Convergence of sensitivity estimates for different Runge Kutta schemes and Bateman sub-cycle levels	217
5.11	Convergence of the absolute QOI error estimate	223
5.12	Convergence of the relative QOI error estimate	224
5.13	Verification of parameter derivative convergence rate for a nuclide in- ventory problem	229
5.14	Verification of parameter derivative convergence rate for a detector response problem	230
6.1	Shuffling schemes for the traveling wave reactor problem	234
6.2	Reference k-eigenvalue solutions for the TWR problem.	235
6.3	Actinide chain for the traveling wave reactor benchmark problem . .	237
6.4	Uncontrolled eigenvalue curves for traveling wave reactor benchmark problem	240
6.5	Comparison of reference and PDT eigenvalue solutions over a single equilibrium cycle	242
6.6	Comparison of PDT and reference profiles of burnup, flux, and power density in the equilibrium cycle.	246
6.7	BOEC and EOEC density profiles for ^{235}U	247
6.8	BOEC and EOEC density profiles nuclides in the ^{239}Pu production chain.	248
6.9	Comparison of equilibrium-cycle uncontrolled eigenvalue for 5 different angular and anisotropy models	249
6.10	Comparison of parameter sensitivity measure for reaction rate QOI .	252
6.11	Predicted vs actual QOI for the traveling wave reactor UQ study . .	254
6.12	Predicted vs actual QOI perturbation for the multi-cycle traveling wave reactor sensitivity test	258
6.13	Grid and material layout for 9-pin BOL calculation.	262
6.14	Spatial variation of gradient of BOL reactivity with respect to ^{10}B concentration (S_2).	265

6.15	Spatial variation of gradient of BOL reactivity with respect to ^{10}B concentration (S_{12})	266
6.16	Spatial variation of thermal flux at initial ^{10}B concentration (S_{12})	267
6.17	Eigenvalue over first year of NSC depletion calculation	271
6.18	Power density profile at beginning of NSC depletion calculation	272
6.19	Comparison of NSC power density across center pin at $t=0$ and $t=1$ year	273
6.20	Uranium-235 number density over 1 year of NSC depletion	274
6.21	Plutonium-239 number density over 1 year of NSC depletion	275
6.22	Samarium-149 number density over 1 year of NSC depletion	276
6.23	Xenon-135 steady-state concentration after 1 year of NSC depletion .	277
6.24	Comparison of ^{135}Xe concentration at 5 days using half-day and 5-day time steps	278
6.25	Comparison of k_{eff} at 5 days using half-day and 5-day time steps	279
6.26	Sensitivity of final reactivity with respect to nuclide initial density	281
6.27	Scheme memory footprint for the 400k unk/cpu problem.	288
6.28	Fixed-source solve count for the checkpoint scheme scaling study	290
6.29	Sweep count for the checkpoint scheme scaling study	292
6.30	Sweeps per fixed-source solve for the checkpoint scheme scaling study	293
6.31	Recovery sweep count for the checkpoint scheme scaling study	295
6.32	Time to solution for the checkpoint scheme scaling study	296
6.33	Weak scaling results for the STOR_ALL and STOR_MOM schemes .	298
6.34	Weak scaling results for the CKPT_ALL and CKPT_MOM schemes .	299
6.35	Scaling the unknown count for the checkpoint scheme scaling study .	301
B.1	Example <component_def> block for depletion in PDT input file	341
B.2	Example <bp_info> input file block for OPERATOR_SPLIT mode	343
B.3	Example component depletion summary	344

B.4	Example <bp_info> input file block for RUNGE-KUTTA mode	346
B.5	Example <QOI_def> block	348
B.6	Example <adjoint_def> block	350
C.1	Convergence of predicted QOI, depletion adjoint problem	398
C.2	Convergence of predicted QOI sensitivity, depletion adjoint problem with scattering	399
C.3	Convergence of the predicted error in the QOI, depletion adjoint problem with scattering	403
F.1	List of nuclides in the 33g ANL fast-spectrum cross section dataset .	414

LIST OF TABLES

TABLE	Page
4.1 Number of forward, adjoint, and recompute fixed source solves for each checkpointing scheme	162
4.2 Single sweep cost and RAM footprint of the checkpointing schemes .	162
5.1 Parameters for infinite medium, 2 group k-eigenvalue depletion verification problem	168
5.2 Problem definition for the 33-group Plutonium-239 production test problem	171
5.3 Reference and PDT terminal densities for the 33g ^{239}Pu production problem	173
5.4 Cross sections for the steady-state, infinite medium adjoint verification problem	177
5.5 Results for infinite medium adjoint verification problem	179
5.6 Parameters for infinite medium, 2 group k-eigenvalue adjoint verification problem	205
5.7 Numerical results for the infinite medium, k-eigenvalue adjoint verification problem	206
5.8 Computed derivatives for the infinite medium, k-eigenvalue adjoint verification problem	207
5.9 Initial densities for the detector response depletion perturbation verification problem	227
6.1 Composition of feed material for traveling wave reactor benchmark .	233
6.2 Tabulated reference equilibrium cycle eigenvalues for the traveling wave benchmark problem (given as ± 0.001)	235
6.3 Specification of problem and solver settings for equilibrium cycle benchmark solution using PDT	238
6.4 Comparison of tabulated equilibrium cycle eigenvalues for the traveling wave benchmark problem (± 0.001)	241

6.5	Global neutronics parameters computed with PDT for the TWR benchmark problem (± 0.001)	243
6.6	Benchmark results for select assemblies in the traveling wave reactor problem	244
6.7	Parameters chosen for ^{239}Pu reaction-rate UQ study	253
6.8	BOL NSC fuel isotopic composition	263
6.9	BOL NSC coolant isotopic composition	263
6.10	Summary of Newton iterations towards target Boron density for NSC BOL problem	268
6.11	Initial fuel material composition for NSC depletion calculation . . .	269
6.12	Finite difference and adjoint-based estimates for reactivity jumps due to 5% initial condition perturbations	282
6.13	Relative sensitivity of nuclear data in NSC simulation	283
6.14	BOL nuclide list and densities for checkpointing scheme scaling prob- lem	285
6.15	Variation in unknowns per processor for scaling study	286
C.1	Parameters for infinite medium, 2 group adjoint verification problem	389
C.2	Infinite medium, 2 group adjoint verification problem results	390
C.3	Parameters for infinite medium, 2 group adjoint verification problem with scattering	396
C.4	Predicted vs. computed explicit Euler result	397
F.1	List of parameters and values for the synthetic, 2-group, 4-component depletion model	413

1. INTRODUCTION

In this work, we describe the development, implementation, and testing of a framework for estimating error and uncertainty in nuclear reactor depletion calculations. The depletion process in a reactor involves the time-dependent production and destruction of nuclides due to fission, absorption, decay, and other interactions with the free neutron field. Likewise, changes in the nuclide populations cause the shape and intensity of the neutron field to evolve over the life cycle of the reactor. Therefore, a depletion calculation requires the simultaneous simulation of the nuclide density field, the neutron field, and the interactions between them. Solutions to these calculations help predict key reactor quantities of interest (QOIs), such as power shape and fuel burnup, and serve as inputs to operational and licensing decisions.

Our framework leverages the mathematical approach known as the adjoint technique. Using this technique, we develop a secondary, or adjoint, system of equations that are related (in a well-defined mathematical sense) to the primary, or forward, depletion equations. As we will demonstrate, the solutions to the forward and adjoint equations can be combined efficiently to produce estimates of the inherent error and uncertainty in predicted QOIs. Obtaining the adjoint solution, however, poses unique computational and algorithmic challenges, especially as the fidelity and complexity of the target problem increase and as advanced computer architectures evolve.

This dissertation describes our framework, which we develop with a general approach in which it is straightforward to add models of coupled physical phenomena. We apply the framework to develop the adjoint equations corresponding to both the source-driven and k-eigenvalue forward depletion formulations. We implement

our framework in the PDT code, a massively-parallel discrete ordinates (S_N) linear transport solver[1, 2]. The implementation is tested using a series of verification problems, including comparisons to analytic and benchmark solutions. We then apply our framework to make QOI predictions (with uncertainty) in larger reactor analysis calculations, and we demonstrate the efficiency of our techniques for executing the forward, adjoint, and UQ calculations on parallel architectures.

The fundamental contribution of this work is the rigorous development, implementation, and testing of an adjoint depletion solver that solves the full neutron transport equation and scales to large, relevant problems. We argue that the adjoint approach is the only viable option for producing uncertainty and error estimates in these large problems, and the novelty in our implementation is in its design for managing the computational challenges associated with high-fidelity solutions on advanced parallel computers.

In the remainder of this introduction, we describe our task in the context of the predictive science and engineering (PS&E) discipline. We discuss the challenges in solving predictive science equations, and we motivate the need for accurate, reliable estimates of error and uncertainty in their solutions. We then provide more background on the advantages and challenges of the adjoint approach. Finally, we highlight the main accomplishments of this research and preview the remaining sections of the dissertation.

1.1 The Challenge of Uncertainty Quantification in Predictive Science Calculations

Predictive science and engineering is a young but rapidly evolving discipline. Its goal is straightforward: to (a) develop a model, often a set of integral or differential equations, that represent a physical process, (b) apply mathematical and computa-

tional techniques to solve the model equations and predict outcomes, or quantities of interest, and (c) produce reliable estimates of the errors and uncertainties in the predictions. Although we focus on scientific models, such as climate change and aircraft design, less obvious applications, such as retail forecasting, stock market analysis, and communications design are increasingly influential in our everyday activities. Decision-makers are relying more and more on PS&E to inform policy, and this trend is increasing with the advancement of computing technology. For the case of high-consequence decisions and policy, the scientific community has undertaken a tremendous effort to make PS&E a reliable tool. This research is in support of that effort.

Uncertainty quantification and sensitivity analysis (UQSA), the general topic of this dissertation, is an important ingredient for the assessment of accuracy and reliability in predictive science calculations[3]. We maintain a broad definition for uncertainty quantification and sensitivity analysis: they are the tasks of (a) recognizing the inherent sources of error in a computational model, and (b) developing rigorous methods for estimating and reporting the manifestation of that error in predictions derived from the model. We discuss several possible sources of error (or uncertainty) in this dissertation, including the effects of the discretization of a continuous model, the propagation of uncertainty in physical properties or other parameters, and the sensitivity of a model to its initial conditions. The effects of these uncertainties will be estimated numerically using the adjoint approach, a mathematical technique that we claim is most effective for producing UQ information in complex computational science calculations.

We also address the processes of verification and validation, which are important parts of any UQSA effort. Verification assesses the accuracy by which the equations that describe a model are solved: are we solving the model equations correctly? The

implementation of a mathematical approach on a computer requires extensive verification to flush out software-related errors (or “bugs”) and to determine the behavior of the implementation in certain regimes. Validation assesses the relevance and/or applicability of a model to the particular prediction of interest: are we solving the correct equations? Validation requires comparison with measured data from previous events or experiments, from which inferences about model error are drawn. It also requires expert judgment to determine whether the model error inferred from previous measurements will be quantitatively similar to the model error for the system and QOIs being predicted. This judgment must be based on an understanding of the simplifying assumptions in the model as they relate to the physical phenomena of the previous measured systems and QOIs compared to the predicted system and QOIs.

An extensive UQSA effort, including verification and validation, provides the substantive, quantitative evidence by which a computational model that simulates real-world physics may be justified as an input to high-consequence decisions. Further, UQSA lends insight into the dynamic behavior of complex systems. Such insight is not available through experimentation alone, especially as multi-scale, multi-physics, and high-dimensional problems become the norm.

Conceptually, it may seem that a completed UQSA study is the final step before a predictive model can be useful for a particular application. In practice, however, UQSA is a continuous effort that often raises several questions for each one it answers. For large, relevant problems, the modeler (developer of the model) and decision maker (user of the model) must consider a large set of factors that may introduce uncertainty and error into predictions, and they must make choices about how to address these factors. These choices, which may be informed by expert judgement and budget/regulatory constraints, impact the validity of the model and therefore

must be evaluated and justified. It is precisely the evaluation of these choices that is an intractable task in full; most often it must be carried out in pieces, which requires further justification.

Recently, the National Academy of Sciences[3] identified a list of sources of error and uncertainty in predictive science calculations. Here, we review this list as it applies to this dissertation:

1. Model validity: The first step in a PS&E study is choosing an appropriate mathematical model for the physics of interest. This choice is driven by the QOI, available resources, and expert judgement. Model discrepancy, or error due to inadequacy of the model itself, is almost always present in complex science and engineering calculations. The reactor analysis community generally believes that error in our model (the neutron transport equation coupled with the nuclide production and depletion equations) is small relative to the requirements of our applications. Thus, in our work, we will not address the validity of the fundamental model equations.
2. Numerical accuracy: This is the accuracy to which the discrete problem is solved. This includes choice of algorithms and tolerances, as well as the detection and elimination of code bugs. As mentioned above, verification is the task of assessing this form of uncertainty, and we devote a section of this dissertation to this effort.
3. Non-linearity of the underlying physics: Non-linearity is defined by the type of interaction between the unknowns in a problem. If this interaction is complex, random, or incompletely understood, simplifying assumptions, linearization techniques, and/or iterative techniques may be required to solve the problem. These strategies introduce errors that will propagate to the predicted QOI. As

we will see, the depletion equations are non-linear.

4. Multi-scale phenomena: Multi-scale problems are those in which the defining characteristics of the different components of the solution evolve (e.g. in time) at different rates. Resolution of the rapidly evolving components requires larger computational costs. The modeler typically decides that features below some threshold scale will not be resolved, introducing error. In our case, we will see that some components of the nuclide field evolve much faster than the neutron field.
5. Level of fidelity: We define fidelity as the accuracy to which the underlying problem is transformed to a discrete system. For example, curved geometries may be represented with straight lines, and the number of line segments used determines the fidelity. Decisions about fidelity are typically driven by the scales of the solution. We address a number of fidelity issues in space, time, energy, and angle, and we develop an estimator for the error introduced by discretization in time.
6. Uncertainty in inputs: Parameters that define the problem may be uncertain. For example, material properties may be difficult to measure and initial/boundary conditions may be inexactly known. Both forms of uncertainty are present in the depletion problems we treat in this dissertation.

The modeler and decision maker must address each of these challenges within the context of the particular problem. They cannot all be addressed at once, and some may only be partially tractable. Thus, the UQSA task will never be complete. It is aided, however, by advancements in computing technology and the extensive research momentum generated by the community in recent years. These tools, along

with the open recognition and documentation of sources of error, work to elevate the credibility of predictive science in policy and decision-making arenas.

The nuclear engineering field, specifically reactor design, is one such arena. Many reactors in the current fleet of US light-water reactors are undergoing power uprate and life extension applications. Both require certification that the systems in the plant can perform above and beyond the original design specifications, which is made possible by more accurate and reliable uncertainty estimates. Moreover, UQSA stands to make a large impact on the design and licensing processes of advanced or next generation reactor designs. Many such designs rely on complex physics to improve economic viability and safety/security performance. Simulations of these plants require high-fidelity models and predictions, along with equally well-resolved uncertainty and sensitivity information.

The most important physical phenomena in a reactor analysis calculation are neutron transport, nuclide production/destruction, heat transfer, fluid flow, and structural mechanics/material performance. These physical phenomena are coupled – each affects the others. Simultaneous simulation of each phenomenon with high fidelity is a formidable challenge and an active area of research. In this dissertation, we present a general, multi-physics-enabled framework for producing UQSA information. For numerical results and analysis, however, we will focus on the interaction between a neutron field and a nuclide density (or composition) field. An accurate model of this interaction will help a reactor analyst balance fuel and poison concentrations. More importantly, the quantification of uncertainties and an understanding of the system's sensitivities helps the analyst choose appropriate margins and improves the basis for high-consequence design decisions.

Another challenge in performing UQSA on high-fidelity problems is the sheer amount of computing horsepower required to complete the calculations. Over sev-

eral decades, the capacity of leading-class machines, often measured in floating point operations per second, or FLOPS, has increased exponentially due in part to machine size (cpu count) and in part to technological advancements and chip design. These capabilities have opened more resources for UQSA experiments by allowing modelers to increase both the quantity and fidelity of their simulations. In other words, increases in computing capacity have enabled the PS&E community to include uncertainty and error estimates in their calculations.

More recently, however, design barriers, such as limits on transistor density, and practical limits, such as cost and power consumption of the new machines, have threatened to flatten the exponential trend in computing capacity. In order to continue to increase FLOP rates, machine designs are forced to increase the cpu count; to control the power requirement, however, new machines have much lower RAM availability per cpu. Today, the FLOP rate goal that the community is targeting is the Exascale, or 10^{18} FLOPs per second, while leading class machines operate at around 10^{16} FLOPs per second. Research towards major hardware technological advancements is underway to bridge this gap.

The trend of higher cpu counts but less available memory has an immediate effect on software design[4, 5]. Traditional algorithms that are less memory-conscious will not scale (or maybe even fit!) on the proposed Exascale architectures. Instead, software designers must pay much closer attention to the memory footprint of their codes. As we will see in the next section and throughout this dissertation, the adjoint technique is traditionally very memory intensive. Towards our goal of thinking about implementations on new and future architectures, we propose and demonstrate a novel technique for reducing the memory footprint of the adjoint technique applied to transport problems.

1.2 The Adjoint Approach for Generating Uncertainty and Error Information

This section provides an overview of the adjoint method, the mathematical technique that we use to generate UQSA information. We give the steps towards the formulation of an adjoint problem, but more importantly, we justify our argument that the adjoint technique is the most viable option for uncertainty and sensitivity studies in large, complex problems.

The distinguishing feature of the adjoint approach is the formulation of a secondary problem, sometimes called the adjoint or dual problem, that is mathematically related to the primary, or forward problem. This adjoint problem is formulated such that its solution can be combined, or cross-correlated, with the forward solution to produce the desired UQSA information. In later sections, we provide the full details of this formulation in a general framework. Here, we provide a flavor for how the adjoint solution is used and for the computational challenges in formulating and solving the adjoint problem.

The adjoint technique is best illustrated with an example problem. Consider a QOI, $Q(x)$, which depends on the solution x to a particular problem. For example, if the problem is to compute a temperature field, the QOI could be the average temperature of that field. Express this forward problem as

$$Ax = b,$$

where A is the operator that we have chosen to describe our problem and b is the source term (e.g., heat source) driving our solution. So, the steps to computing Q would be to (a) invert operator A , which gives x , and (b) compute $Q(x)$.

Now consider the same problem, but define p as a set of parameters that define the system. For our example temperature problem, these could be material heat

capacities or boundary temperatures. In this case, our forward problem is to find $Q(x, p)$, where

$$A(p)x = b(p).$$

Suppose that our UQSA task is to compute the variation, or sensitivity, of Q with respect to perturbations in each parameter p :

$$\frac{dQ}{dp} = \frac{\partial Q}{\partial p} + \frac{\partial Q}{\partial x} \frac{dx}{dp}.$$

We cannot evaluate this derivative directly because x depends on p in an unknown way. We could estimate $\frac{dx}{dp}$ with a finite difference approximation:

$$\frac{dx}{dp} \approx \frac{x_1 - x_0}{\delta p},$$

where x_0 is the unperturbed solution satisfying $A(p_0)x_0 = b(p_0)$, and x_1 is the perturbed solution, $A(p_0 + \delta p)x_1 = b(p_0 + \delta p)$. Note that this strategy requires one inversion of the operator A for *each* perturbation in p ; moreover, the estimated sensitivities are subject to a first-order error in δp , the magnitude of which may be difficult to estimate.

The adjoint approach is an alternative to the finite difference approach. It defines an adjoint operator A^\dagger and adjoint source b^\dagger that form the adjoint problem

$$A^\dagger(p)x^\dagger = b^\dagger(p).$$

The operator A^\dagger typically has the same flavor as the forward operator A and therefore carries a similar cost to invert; the source term b^\dagger is derived from the form of the QOI. The relationship of both operators to their counterpart in the forward problem

depends on the characteristics of the problem and the mathematical techniques used in the model. If we can invert A^\dagger to produce x^\dagger , then our QOI sensitivity can be expressed as

$$\frac{dQ}{dp} = \frac{\partial Q}{\partial p} - x^\dagger \frac{\partial Ax}{\partial p} + x^\dagger \frac{\partial b}{\partial p}.$$

We can evaluate each of these terms, as they are simply partial derivatives. This expression requires both x and x^\dagger , so the computational cost is on the order of two inversions of A .

Here is the crucial point: this sensitivity expression holds for *every* p in the problem. Whereas the finite difference estimator required an extra inversion of A for each p , the adjoint approach requires exactly one forward solve and one adjoint solve, regardless of the size of p . To say it another way, the cost to produce our desired UQ information using the adjoint approach is on the order of the cost of two forward solves no matter the number of uncertain parameters.

In a nuclear engineering problem, for example, the parameter vector may contain all cross sections for all materials, decay constants, fission yields, branching ratios, initial conditions, etc. It is not difficult to imagine a parameter vector with tens of thousands of entries. The same dimensionality is found in most physical problems of importance. This is why the adjoint approach is attractive: it provides UQSA information at a fixed computational cost. As we will see, the mathematics behind the approach are not based on approximations or assumptions, as in the finite difference approach, providing a rich network of research and theory. We will also show that the adjoint solution provides a way to estimate error in addition to sensitivity, again at a fixed cost. Thus, the adjoint technique is an attractive UQSA approach for our problems of interest.

From a computational perspective, the adjoint approach carries a unique set of

challenges, some of which may be illustrated in our example problem. First, the adjoint problem typically carries the same complexities as the forward problem. Formulation of the adjoint operator A^\dagger may be challenging in a multi-physics environment, a task we facilitate in this dissertation using a very general approach. Also, the adjoint operator may also be difficult to solve, requiring additional or more specialized numerical methods and software. This is certainly true in this dissertation, as much additional software is required to form the operator Jacobians and partial derivatives that appear in the adjoint and sensitivity equations.

The second and most formidable challenge in solving the adjoint equations is a matter of logistics: the forward solution is required to form the operators and source terms in the adjoint equation. We will see that A^\dagger is often a linearization about Ax and therefore contains x in general. Similarly, b^\dagger may contain x . The sensitivity equation certainly requires both x and x^\dagger . For small problems, making the forward solution available during the adjoint solve is not a memory burden on the host machine. For problems with millions of unknowns, perhaps at different time steps, simply storing the forward solution in memory is not an option. Instead, schemes are available for “checkpointing” and recomputing the forward solution as necessary.

Schemes for managing access to the forward solution require considerable software development (and the verification and testing that comes along with it). We devote a significant amount of this dissertation to our implementation of these schemes. In particular, we develop and demonstrate a family of schemes that take advantage of a dimensionality reduction in the forward solution. This results in large efficiency gains over traditional algorithms for large problems on large numbers of processors, making the adjoint approach a feasible option for depletion problems using the full transport equation.

To summarize, the adjoint approach can provide sensitivity and error information at a computational cost that does not scale according to the number of uncertain dimensions. It is based on sound mathematical techniques and has been applied in many fields for a number of years. Its implementation may require significant software development and testing; hence the subject of this dissertation. For dedicated UQSA studies on problems with many unknowns and many uncertain parameters, however, we argue that this development cost is outweighed by the theoretical advantages and efficiency of the adjoint approach.

1.3 Main Contributions of this Research

We faced a number of theoretical and computational challenges through the course of this research. Our solutions and strategies for attacking these challenges led to a number of major accomplishments and contributions to the science community. We highlight these accomplishments as follows:

- We overview a framework for deriving a system of adjoint equations corresponding to a general system of forward, engineering equations. This abstract formalism, which appears more frequently in mathematics communities than in the nuclear engineering community, provides the form of the adjoint equations as well as expressions for estimating QOI sensitivity and numerical error estimates.
- We apply our framework to the equations describing depletion in an operating nuclear reactor and show that the resulting equations are identical to those adjoint equations developed in earlier nuclear engineering literature.
- We focus on solving the forward and adjoint depletion equations using the fully angular dependent transport equation, and doing so efficiently for high-fidelity

problems on large processor counts. Transport-depletion solutions with adjoint functionality are not widely reported in the literature to date because they pose large computational challenges.

- We describe and implement a family of checkpointing schemes that can drastically reduce the memory and I/O load on the host machine. Our simple scaling study shows that the new schemes do decrease memory and I/O costs without increasing the overall time to solution significantly. Our conclusion is that high-fidelity problems on memory-limited architectures must use some variant of our checkpointing strategy to achieve efficient scaling performance.
- We apply our depletion-perturbation framework to a traveling wave benchmark problem and to simulations of the Nuclear Science Center reactor at Texas A&M. Results indicate that the uncertainty quantification information produced in these simulations is accurate and will be useful to ongoing research efforts within the department.

1.4 A Preview of the Remaining Sections

In this section, we posed the depletion problem within the larger context of predictive science and engineering. We discussed the need for detailed, rigorous uncertainty quantification and sensitivity analysis in order to justify the validity of computational results, and we described the formidable challenges facing a modeler tasked with UQSA for large, realistic problems. We then introduced the adjoint technique, illustrated its efficiencies, and discussed the challenges in its implementation.

In Sec. 2, we cover a range mathematical and computational preliminaries that will serve as the foundation to our adjoint formalism and its implementation in the PDT code. Section 3 gives the adjoint formalism in full detail. We begin the section

with a derivation of a completely general set of adjoint equations, and we show exactly how the adjoint variable is used to produce both QOI sensitivity and error estimates. We then apply the formalism to both the source-driven and k-eigenvalue formulation of the depletion equations and give the specific adjoint equations for each case.

In Sec. 4, we provide an overview of the implementation of the adjoint formalism in the PDT code. This includes a description of the solvers and classes that perform the adjoint time integration. This section is also where we introduce our new checkpointing schemes, complete with memory footprint and computational cost analysis. In Sec. 5, we describe and give examples from our verification strategy. Our approach is to verify in a hierarchical manner, wherein the building blocks are verified independently first, and then combined in a series of increasingly complex problems.

Section 6 describes the application of our adjoint formalism to two larger, relevant reactor analysis problems. The first is a benchmark problem simulating a one-dimensional traveling wave reactor. We find that our results to the forward problem are consistent with that of other codes, and we enhance the benchmark problem by simulating a number of UQ studies using our adjoint capability. The second problem is related to an effort to simulate the NSC-TRIGA reactor at Texas A&M University. Here we use our adjoint capabilities to calibrate beginning-of-life configurations. The last section of Sec. 6 describes a scaling study for testing the family of checkpointing schemes as both problem size and processor count are increased. Our efficient checkpointing schemes outperform the traditional schemes.

Finally, in Sec. 7, we provide concluding remarks and recommendations for further research.

2. PRELIMINARIES

This section covers theoretical foundations and computational techniques upon which the remainder of the dissertation will build. In the first section, we introduce the neutron transport and Bateman equations, which together compose the depletion equations, and give both the source-driven and k-eigenvalue forms of the forward depletion problem. In Sec. 2.2, we show that the depletion equations fall into the general class of differential-algebraic equations.

In Sec. 2.3, we introduce the Runge-Kutta time discretization method, one of two methods we use for marching the unknowns through time. This section describes permutations of the Runge-Kutta schemes, including those that provide error estimates for uncertainty propagation. In Sec. 2.4, we give the form of the adjoint transport and Bateman operators that we will employ throughout the dissertation. We also give a more detailed example problem that motivates the use of the adjoint approach.

Sections 2.5 and 2.6 review the relevant theoretical and computational work available in the literature. In the former, we discuss the issue of adjoint consistency for both spatial and temporal discretization. We make a slight extension of the temporal piece to account for our differential-algebraic system. In the latter, we review literature in from the nuclear engineering community in the fields of perturbation and depletion perturbation theory. Although the community has a rich history in the application of adjoint technology, advances in recent years have been less frequent, and very few reports include high fidelity, full transport solutions.

2.1 The Depletion Equations

The depletion equations describe the interaction of free neutrons with the nuclei of the materials in a nuclear reactor. Free neutrons are those that are not bound to a nucleus; each moves with a certain speed and direction among the nuclei around them. Eventually, these neutrons either escape the reactor or interact with a nucleus by either an absorption or scattering process. An absorption reaction may be accompanied by the emission of other particles from the nucleus and results in a change in the number of protons (Z) and or neutrons (A) in the nucleus. A scattering reaction does not result in change of A or Z , but may change the speed and direction of the neutron. A depletion calculation keeps track of these interactions over some operational cycle and models the resulting changes in the free neutron concentration and material densities in the reactor.

The neutron field is modeled by the transport equation, which we write in operator form as

$$\frac{1}{v} \frac{\partial \psi}{\partial t} + H(N, p, t)\psi = G(N, p, t)\psi + S(p, t). \quad (2.1)$$

The angular neutron flux, $\psi(r, E, \Omega, t)$, is the speed, $v(E)$, times the number of neutrons at time t and spatial location r with energy E moving in direction Ω , per unit of volume, energy, and solid angle. The operators H and G depend on N , the list of unknown nuclide densities, and p , a set of parameters or inputs that define or are required to solve the system. The term $S(p)$ is included for completeness and represents a prescribed volumetric neutron source, possibly varying in time, space, energy, and angle. We have not included the contribution of delayed neutrons because, during approximately steady state reactor operation, the delayed-neutron source can be included in the fission term ($G\psi$) with negligible error.

The operator H contains an advection term, a collision term, and a scattering term, written as

$$H(N, p, t)\psi \equiv \Omega \cdot \nabla \psi(r, E, \Omega, t) + \Sigma_t(N, E, t)\psi(r, E, \Omega, t) - \int dE' \int d\Omega' \Sigma_s(N, E' \rightarrow E, \Omega' \rightarrow \Omega, t)\psi(r, E', \Omega', t).$$

Here, Σ_t and Σ_s are the (spatially varying) macroscopic total and scattering cross sections, respectively, defined as

$$\Sigma_t(N, E, t) = \sum_{k=1}^K N_k(r, t) \sigma_{t,k}(E)$$

$$\Sigma_s(N, E' \rightarrow E, \Omega' \rightarrow \Omega, t) = \sum_{k=1}^K N_k(r, t) \sigma_{s,k}(E' \rightarrow E, \Omega' \rightarrow \Omega)$$

where K is the number of nuclides present at r and σ_k is the microscopic cross section of nuclide k . The operator G is the fission source term, written as

$$G(N, p, t)\psi \equiv \frac{\chi(N, E, t)}{4\pi} \int dE' \nu \Sigma_f(N, E', t) \int d\Omega' \psi(r, E', \Omega', t).$$

Here $\chi(E)$ is the energy spectrum of fission neutrons. In a mixture of materials, we adopt a number density weighted approximation for χ , namely

$$\chi(N, E, t) = \frac{\sum_{k=1}^K N_k(r, t) \chi_k(E)}{\sum_{k=1}^K N_k(r, t)}.$$

This approximation affords computational savings over the exact representation, which is fission-production weighted. Finally, the term $\nu \Sigma_f$ is the macroscopic fission neutron production cross section, defined in terms of the local isotopic composition

as

$$\nu\Sigma_f(N, E, t) = \sum_{k=1}^K N_k(r, t)\nu_k(E)\sigma_{f,k}(E).$$

There are two important steady-state versions of Eq. (2.1). The source-driven form is

$$H(N, p, t)\psi - G(N, p, t)\psi = S(p, t),$$

which may be appropriate for subcritical reactor or shielding calculations. The k-eigenvalue form is

$$H(N, p, t)\psi = \lambda(t)G(N, p, t)\psi,$$

where $\lambda = \frac{1}{k_{\text{eff}}}$, and k_{eff} is the multiplication factor of the system. For reactor calculations, this formulation must be accompanied by a power constraint to specify the magnitude of the flux. This is the proper model for the simulation of the usual approximately steady-state reactor operation.

The Bateman equations, which describe the growth, decay, and transmutation of the nuclide densities are written simply as

$$\frac{dN}{dt} = B(\psi, p, t)N \quad (2.2)$$

where again N is the list of unknown nuclide densities, which vary as a function of space, and B is an operator that describes mechanisms by which these densities change (i.e. absorption, decay, etc.). In our model, there is a set of equations (2.2) at each spatial point in the reactor. Each of these sets of equations are independent because we do not model spatial migration of nuclides (as may happen with gaseous fission products, for example). The Bateman operator at a given position depends on the neutron flux and the parameter vector at that position. We treat the variable N as a list of elements in the reactor, each of which varies as a function of space.

Equations (2.1) and (2.2), the fully time-dependent nonlinear burnup equations, have been the subject of extensive effort in theory and software development over the last 50 years[6, 7, 8]. Because the neutron flux generally changes on time scales much longer than the time scales of some nuclide densities (the system is stiff), nearly all solution schemes employ some variant of the quasi-static approximation[9]. In this approximation, the neutron flux is decomposed into a shape and amplitude function. The shape function is obtained using a steady-state version of Eq. (2.1); here, we employ the k-eigenvalue form of the equation, which is most relevant to reactor design and analysis. The change in the flux amplitude in time over each cycle is assumed to follow a particular functional dependence. Early work assumed that the amplitude function was constant over each in a series of time steps[10, 11]. Later work generalized this idea to allow for an amplitude function that is linear in time[12].

The quasi-static, k-eigenvalue form of the burnup equations is

$$\frac{dN}{dt} = B(\psi, p)N \quad (2.3)$$

$$H(N, p)\psi = \frac{1}{k_{\text{eff}}}G(N, p)\psi \quad (2.4)$$

$$A(t)P(N, p, t)\psi - \mathcal{P}(t) = 0, \quad (2.5)$$

where Eq. (2.5) is a normalization of the flux magnitude to a prescribed power level \mathcal{P} . Unknown A is the normalization factor and the operator P is an integration of the energy produced via fission (we ignore energy production via capture and other reactions):

$$P(N, p, t) = \int dV \int dE \int d\Omega \psi(r, E, \Omega) \sum_{k=1}^K N_k(r, t) E_{f,k} \sigma_{f,k}(E),$$

where $E_{f,k}$ is the energy per fission for nuclide k .

Formally, Eqs. (2.3)–(2.5) constitute a system of differential-algebraic equations (DAEs), as one subset of the unknowns (N) satisfies time-differential equations while the others (ψ) satisfy a constraint that is algebraic in time. DAEs will be discussed in more detail in the following section. The UQSA problem is to solve this system for reactor quantities of interest (QOIs) and to provide some estimate for the error in the QOIs as well as the sensitivity of the QOIs with respect to the individual parameters in p .

The depletion equations, (2.3) and (2.4), are characterized by a high-dimensional p . For example, in a modest calculation tracking 20 nuclides, 50 energy groups, and 100 weighting spectra, the number of cross sections required as inputs to the problem will number over one million! If our task is to compute the sensitivity of a handful of QOIs with respect to a large number of parameters, we cannot rely on finite-difference or brute-force sampling methods because of the prohibitively large number of code runs that would be required to cover the high-dimensional space of p .

This data dimensionality challenge motivates the use of adjoint-based methods, which can provide sensitivity information at the cost of just a few forward solves per QOI[13]. Further, the adjoint formalism provides expressions for propagating local truncation error estimates into global QOI error estimates, which should be more accurate and efficient than error-extrapolation methods. We provide an introduction to adjoint methods in Sec. 2.4.1 and develop a general, multi-physics enabled adjoint framework in Sec. 3.1.

2.2 Differential-Algebraic Equations

We typically model the behavior of physical systems using differential equations. Ordinary or partial differential equations are perhaps the most familiar formulation, where a set of coupled equations, each of which contain a time derivative, describe the trajectory of a solution vector given an initial condition. Many physical and engineering systems, however, involve constraints on the solution that do not appear in time-differential form. For example, consider the movement of point masses that are constrained to a geometric surface. The equations of motion would describe the movement of the masses; an algebraic constraint would confine the movement to the surface.

Systems consisting of both differential and algebraic equations are called differential-algebraic systems or systems of differential-algebraic equations (DAEs). We will encounter DAEs throughout this dissertation. The DAE system can be written most generally as[14]

$$F(\dot{x}, x, t) = 0, \quad x(t_0) = x_0,$$

where x is the solution and \dot{x} is its derivative with respect to time. We find it more useful to discuss the DAE in terms of its differential and algebraic components, which we denote with superscripts d and a , respectively. The expanded version is

$$\begin{aligned} F(\dot{x}, x, t) &= \begin{Bmatrix} F^d(\dot{x}, x, t) \\ F^a(\dot{x}, x, t) \end{Bmatrix} = \begin{Bmatrix} \dot{x}^d - f^d(\dot{x}, x, t) \\ f^a(\dot{x}, x, t) \end{Bmatrix} = 0 \\ &x(t_0) = x_0. \end{aligned} \tag{2.6}$$

Here we have assumed a semi-explicit form, where the algebraic constraints are separated explicitly from the differential equations. The differential variables, x^d , are

those whose time derivatives appear in the governing equations; the algebraic variables, x^a , will vary in time, but their time derivatives do not appear explicitly as terms in the governing equations. We will use this notation throughout the derivations in this research.

Similar to ODEs and PDEs, the community has developed a range of techniques for integrating the DAE system in time. We will integrate the DAE system using half-explicit Runge-Kutta methods, described in the following section. The solvability of a system depends on its differentiation index, often defined as the number of times that the algebraic constraint must be differentiated with respect to t in order to write an explicit function for \dot{x}^a in terms of x and t . Index-1 systems involve fairly intuitive solution techniques, while higher-order systems require more complex analysis. For this dissertation, we will only be concerned with index-1 DAEs. In particular, the index-1 property ensures that $\frac{\partial F^a}{\partial x^a}$ is invertible, and we will assume that $\frac{\partial F^a}{\partial \dot{x}} = 0$ and $\frac{\partial F^d}{\partial \dot{x}^d} = I$.

Take the k-eigenvalue burnup equations, Eqs. (2.3) and (2.4), as an example. Indeed $\frac{\partial F^a}{\partial x^a}$ is invertible, as we can solve the k-eigenvalue transport equation (with normalization) uniquely for ψ , no time derivatives appear in the transport equation ($\frac{\partial F^a}{\partial \dot{x}} = 0$), and only the time derivative for density unknown i appears in Bateman equation i , giving $\frac{\partial F^d}{\partial \dot{x}^d} = I$.

2.3 Runge-Kutta Time Discretization Methods

This section introduces the Runge-Kutta method, which is a family of time discretization procedures for solving the general initial value problem

$$\begin{cases} y' = f(t, y), & t \in [t_0, t_f] \\ y(t_0) = y_0 \end{cases} \quad (2.7)$$

for the unknown function $y(\cdot)$. These methods imitate a general Taylor series method to achieve higher-order accuracy in the integration of problem (2.7) by using clever combinations of the steady-state residual $f(t, y)$ as opposed to the burdensome task of successive differentiation of $f(t, y)$ [15]. The general form of an s -stage Runge-Kutta method over time step $t \in [t_n, t_{n+1}]$, $t_{n+1} = t_n + h$, is

$$y_{t_{n+1}} = y_{t_n} + h \sum_{i=1}^s b_i f(t_i, y_i) \quad (2.8)$$

where

$$\begin{aligned} t_i &= t_n + c_i h \\ y_i &= y_{t_n} + h \sum_{j=1}^J a_{ij} f(t_j, y_j), \end{aligned} \quad (2.9)$$

and the coefficients b_i , c_i , and a_{ij} define the particular Runge-Kutta method. These coefficients are often summarized in a Butcher tableau:

c_1	a_{11}				
c_2	a_{21}	a_{22}			
\vdots	\vdots	\vdots	\ddots		
c_s	a_{s1}	a_{s2}	\dots	a_{ss}	
	b_1	b_2	\dots	b_s	

If the diagonal entries in the a_{ij} matrix are zero (or, equivalently, if $J < i$ in Eq. (2.9)), the method is called explicit because the computation of the stage vector, y_i depends only on stage vectors, y_j , $j < i$. The simplest explicit Runge-Kutta method is the familiar forward (explicit) Euler method. Its Butcher tableau is

0	0
	1

which simply results in

$$y_1 = y_{t_n}, \quad \rightarrow \quad y_{t_n+1} = y_{t_n} + hf(t_n, y_1).$$

A very common explicit Runge-Kutta method is the 4th order rule, RK4, with

Butcher tableau

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6.

In the case that the steady state residual is a function of t only (i.e. problem (2.7) is a simple integral), the RK4 method is equivalent to Simpson's numerical integration scheme.

Implicit Runge-Kutta methods ($a_{ii} \neq 0$) may offer improved stability at the cost of a linear or possibly nonlinear solve for the stage vector y_i at *each* stage. If the function f is linear in y , that is $f(t, y) = A(t)y$, then the following linear system must be solved at each stage:

$$\left[I - a_{ii}hA(t_i) \right] y_i = y_{t_n} + h \sum_{j=1}^{i-1} a_{ij}f(t_j, y_j).$$

In the case that $f(t, y)$ is nonlinear in y , each stage requires a Newton (or other nonlinear) solve for the stage vector. This may drastically increase the required number of function evaluations of $f(\cdot)$. Thus, the computational cost of evaluating the steady state residual often limits the choice of implicit Runge-Kutta methods that may be applied to expensive, nonlinear systems. The simplest implicit Runge-Kutta method is the backward Euler method, with Butcher tableau

1	1	
		1

Another familiar implicit Runge-Kutta method is the trapezoidal rule. Its Butcher tableau is

0	0
1	1/2 1/2
,	
	1/2 1/2

which results in the equations

$$\begin{aligned} y_1 &= y_{t_n} \\ y_2 - \frac{h}{2} f(t_{n+1}, y_2) &= y_{t_n} + \frac{h}{2} f(t_n, y_1) \end{aligned} \quad \rightarrow \quad y_{t_{n+1}} = y_n + \frac{h}{2} \left(f(t_n, y_1) + f(t_{n+1}, y_2) \right).$$

Embedded Runge-Kutta methods offer an efficient scheme for estimating local truncation error in the time integration by producing two estimates for the solution at the end of the time step: $y_{t_{n+1}}$, which is of order accuracy p , and $\hat{y}_{t_{n+1}}$, which is of order $(p + 1)$. The Butcher tableau for an embedded Runge-Kutta scheme is written as

c_1	a_{11}			
c_2	a_{21}	a_{22}		
\vdots	\vdots	\vdots	\ddots	
c_s				
	a_{s1}	a_{s2}	\dots	a_{ss}
	b_1	b_2	\dots	b_s
	\hat{b}_1	\hat{b}_2	\dots	\hat{b}_s

The key is that the lower precision estimate, y , is generated with the same set of c_i and a_{ij} coefficients that were used to compute \hat{y} , requiring no extra evaluations of $f(\cdot)$. Only the stage weights, b_i , are different.

If the order p approximation is used to continue the integration, then the local truncation error may be estimated to $\mathcal{O}(h^{p+2})$ accuracy by

$$e_{t_n+1} \approx y_{t_n+1} - \hat{y}_{t_n+1}.$$

It is possible to use this estimate in an adaptive time scheme, as follows. Define a tolerance for the error in component k of the solution as

$$\tau_k \equiv A_k + \max[y_{0,k}, y_k] \cdot R_k$$

where A_k and R_k are the desired absolute and relative tolerances. We can then compute some kind of integral error norm, possibly

$$\|e\| = \sqrt{\frac{1}{N} \sum_{k=1}^N \left(\frac{e_k}{\tau_k}\right)^2}.$$

If we define an optimal time step size, h_{opt} , as the time step for which $\|e\| = 1$ (in some sense, each component achieves its tolerance exactly), and write the error behavior as $\|e\| = C \cdot h^{p+1}$, then a ratio of the current and optimal error leads to an approximation for the optimal time step:

$$h_{opt} = h \left(\frac{1}{\|e\|} \right)^{\frac{1}{p+1}}.$$

In practice, a safety factor may be applied to this equation in order to increase the acceptance rate of new time steps and to prevent the time step size from changing too rapidly. This procedure is repeated until the error is below the desired tolerance. Once this occurs, the local truncation error e_{t_n+1} is retained for later use[16]. Alternatively, one can move forward with the integration using the order- $(p+1)$ accurate

solution. In this case, there is no estimate of the local truncation error, but the difference $y - \hat{y}$ may still be used for time step control.

We employ a slight modification of the explicit Runge-Kutta scheme to integrate the DAE system, Eq. (2.6). These schemes are called half-explicit or half-implicit Runge-Kutta schemes[17]. For example, a half-explicit scheme would integrate the depletion equations, Eqs. (2.3) and (2.4), over time step $t \in [t_{n-1}, t_n]$ as follows:

$$\begin{aligned} N_n &= N_{n-1} + h \sum_{i=1}^s b_i B_i N_i \\ H_i \psi_i - \lambda_i G_i \psi_i &= 0 \\ N_i &= N_{n-1} + h \sum_{j=1}^{i-1} a_{ij} B_j N_j. \end{aligned} \tag{2.10}$$

Here operator and unknown subscripts i and j indicate evaluation at time t_i and t_j , respectively. These schemes are called half-explicit because the differential variables are advanced using explicit time-stepping, but the algebraic constraint is satisfied at each stage.

Alternatively, a half-implicit scheme would advance the depletion equations as

$$\begin{aligned} N_n &= N_{n-1} + h \sum_{i=1}^s b_i B_i N_i \\ H_i \psi_i - \lambda_i G_i \psi_i &= 0 \\ \left[I - a_{ii} h B_{i-1} \right] N_i &= N_{n-1} + h \sum_{j=1}^{i-1} a_{ij} B_j N_j. \end{aligned} \tag{2.11}$$

Here, the differential variable is advanced *implicitly*, but the operator to be inverted depends on the explicit algebraic variable. The half-explicit schemes maintain the same order of accuracy for index-1 DAEs as they would for ODEs, while the half-

implicit schemes may converge sub-optimally because the algebraic variable is lagged one stage behind[18].

A modification of the first-order version of these schemes allows for Bateman sub-cycling, or the advancement of the Bateman equations over each stage on a time-scale shorter than h . Denote this time step as \hat{h} and assume $\hat{n} = \frac{h}{\hat{h}}$ is an integer. Then the first-order, sub-cycled semi-explicit scheme is

$$\begin{aligned} N_n &= N_{n-1} + h[BN] \\ H_n \psi_n - \lambda_n G_n \psi_n &= 0 \\ N_k &= N_{k-1} + \hat{h} B_n N_{k-1}, \quad k = 1 \dots \hat{n}, \quad N_0 = N_n \\ [BN] &= \frac{1}{\hat{n}} \sum_{k=1}^{\hat{n}} B_n N_k. \end{aligned} \tag{2.12}$$

The corresponding first order sub-cycled semi-implicit scheme is

$$\begin{aligned} N_n &= N_{n-1} + h[BN] \\ H_n \psi_n - \lambda_n G_n \psi_n &= 0 \\ [I - \hat{h} B_n] N_k &= N_{k-1}, \quad k = 1 \dots \hat{n}, \quad N_0 = N_n \\ [BN] &= \frac{1}{\hat{n}} \sum_{k=1}^{\hat{n}} B_n N_k. \end{aligned} \tag{2.13}$$

These schemes provide the capability for shorter Bateman time steps. They are also self-adjoint, per the conditions we will develop in the next section, so the same scheme will be used to integrate the adjoint equations backwards in time. As for accuracy, they are limited to first order; they can be interpreted as a midpoint rule for the Bateman variables (the derivative is averaged over the time step), but remain a first order rule for the flux variables. Higher order schemes with sub-cycling will

converge sub-optimally for this same reason. We do note, however, that in the limit of a constant flux, these schemes will achieve second order convergence.

We find that the first order semi-implicit scheme with Bateman sub-cycling is most appropriate for the depletion equations because some components of the Bateman solution (e.g. fission products or short-lived nuclides) tend to evolve on time scales much shorter than the flux variable. State-of-the-art large scale power reactor analysis problems may track 300-400 isotopes, many with very short half-lives, requiring implicit schemes and/or flexibility for very short Bateman time steps. The flux shape and magnitude, however, tends to change on much longer time-scales, especially during steady-state operation. Thus, we find the semi-implicit scheme with Bateman sub-cycling to be the most attractive for the larger simulations.

2.4 The Adjoint Bateman and Neutron Transport Operators

In this section we discuss the adjoint operators corresponding to forward operators that appear in the depletion equations. We first walk through an example adjoint problem to illustrate the value in the approach. We then derive the adjoint transport and fission operators and show that the adjoint transport equation can be solved using the same machinery that solves the forward problem. We end with a discussion of the adjoint Bateman operator, which we write in a particular form that is consistent with the dimensionality of the transport and fission operators.

The transport operator, $H - G$, is a linear and real-valued function; therefore, it has an adjoint operator, $H^\dagger - G^\dagger$, which is defined with respect to a particular inner product over the space on which $H - G$ operates. Similarly, the Bateman equation is linear and therefore has a well defined adjoint operator. The inner product that defines these adjoint operators will be an integral over independent variables, and we will use bracket notation, $\langle \rangle_P$, to denote integration over the variables P . Although we do not treat nonlinear operators in detail in this dissertation, we do note that there is no guarantee of existence of an operator that is adjoint to a forward, nonlinear operator. The conditions under which one or more adjoint operators may exist is described in more detail in Sec. 3.4.

2.4.1 An example problem to motivate the adjoint approach

We begin by re-writing the continuous neutron transport and fission operators:

$$\begin{aligned} H\psi - G\psi &= \Omega \cdot \nabla\psi(r, E, \Omega) + \Sigma_t(r, E)\psi(r, E, \Omega) \\ &\quad - \int dE' \int d\Omega' \Sigma_s(r, E' \rightarrow E, \Omega' \cdot \Omega) \psi(r, E', \Omega') \\ &\quad - \frac{\chi(E)}{4\pi} \int dE' \nu \Sigma_f(r, E') \int d\Omega' \psi(r, E', \Omega'), \end{aligned} \quad (2.14)$$

Unless otherwise noted, each integral is over the full range of its independent variable. We may wish to invert this operator in order to solve for the neutron flux as a function of angle, space, and energy resulting from a fixed source, $S_0(r, E, \Omega)$, in a subcritical system. The problem to be solved is

$$\begin{cases} H\psi - G\psi = S_0 & r \in \mathcal{D} \\ \psi = 0 & \text{on } \left\{ r \in \partial\mathcal{D} \mid \Omega \cdot \hat{n} < 0 \right\} \equiv \Gamma_- \end{cases}. \quad (2.15)$$

Suppose further that we wish to operate on the solution of this problem to compute some quantity of interest (or QOI, or response, or metric) that is of particular interest to our system. We'll denote the QOI with \mathcal{R} and assume that it may be written as an integral of ψ times some function \mathcal{L} over phase space

$$\begin{aligned} \mathcal{R} &= \int dr \int dE \int d\Omega \mathcal{L}(r, E, \Omega) \psi(r, E, \Omega) \\ &= \langle \mathcal{L}\psi \rangle_{E, \mathcal{D}, \Omega}. \end{aligned} \quad (2.16)$$

Note that the function \mathcal{L} may be piece-wise defined to specify the QOI. For example, if \mathcal{R} is fast-group reaction rate in a sub volume of the detector (a fission chamber response, possibly), the function \mathcal{L} may be defined as

$$\mathcal{L} = \begin{cases} \Sigma_f(r, E) & r \text{ in detector volume, } E > E_D \\ 0 & \text{otherwise} \end{cases}$$

Now consider the optimization problem of searching for the source configuration (that is, distribution of S_0), such that \mathcal{R} is minimized or maximized. For each realization of S_0 , problem 2.15 must be re-solved, potentially a costly (in terms of computer time) undertaking. We will now show that the adjoint is particularly well

suited for this kind of problem, as it requires just a single transport solve to solve for all possible \mathcal{R} resulting from any source configuration.

The adjoint equation will be derived using a variational-like approach[19] in preparation for a more traditional variational derivation in a later section. We begin by defining an inner product over all the independent variables or phase space (space, energy, and angle)

$$\langle f, g \rangle_{E,\mathcal{D},\Omega} \equiv \int_{\mathcal{D}} dr \int_0^\infty dE \int_{4\pi} d\Omega f(r, E, \Omega) g(r, E, \Omega).$$

Adjoint operators, H^\dagger and G^\dagger are defined such that they satisfy the following expression involving our inner product:

$$\langle \psi^\dagger, H\psi \rangle_{E,\mathcal{D},\Omega} = \langle \psi, H^\dagger\psi^\dagger \rangle_{E,\mathcal{D},\Omega}, \quad (2.17)$$

$$\langle \psi^\dagger, G\psi \rangle_{E,\mathcal{D},\Omega} = \langle \psi, G^\dagger\psi^\dagger \rangle_{E,\mathcal{D},\Omega}. \quad (2.18)$$

These equations must hold for all functions ψ and ψ^\dagger that live in the correct space for transport solutions in the given problem. Next we take the inner product of ψ^\dagger with a re-arrangement of Eq. (2.15) to write

$$\langle \psi^\dagger, H\psi - G\psi - S_0 \rangle_{E,\mathcal{D},\Omega} = 0.$$

We subtract this quantity from the right hand side of our QOI equation and substi-

tute Eqs. (2.17) and (2.18) as follows

$$\begin{aligned}\mathcal{R} &= \left\langle \mathcal{L}\psi \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, H\psi - G\psi - S_0 \right\rangle_{E,\mathcal{D},\Omega} \\ &= \left\langle \mathcal{L}\psi \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, H\psi \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \psi^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \psi^\dagger, S_0 \right\rangle_{E,\mathcal{D},\Omega} \quad (2.19)\end{aligned}$$

$$\begin{aligned}&= \left\langle \mathcal{L}\psi \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi, H^\dagger\psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \psi, G^\dagger\psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \psi^\dagger, S_0 \right\rangle_{E,\mathcal{D},\Omega} \\ &= \left\langle \psi, \mathcal{L} - H^\dagger\psi^\dagger + G^\dagger\psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \psi^\dagger, S_0 \right\rangle_{E,\mathcal{D},\Omega}. \quad (2.20)\end{aligned}$$

Equation (2.20) reveals that if $H^\dagger\psi^\dagger - G^\dagger\psi^\dagger = \mathcal{L}$, then $\mathcal{R} = \left\langle \psi^\dagger, S_0 \right\rangle_{E,\mathcal{D},\Omega}$. In other words, with one inversion of the (yet-to-be-defined) adjoint operators to obtain the adjoint flux, the QOI may be computed *exactly* for any source configuration with just one integral over the phase space!

With a little more work, we can use the adjoint variable to derive UQ information. Say we are interested in the sensitivity of \mathcal{R} with respect to a list of parameters, p , which, for example, may contain all the group-wise microscopic total, fission, and scattering cross sections. Mathematically, we define sensitivity as a total derivative, $\frac{d\mathcal{R}}{dp}$. We begin by applying this derivative (using the chain rule) to Eq. (2.19) and

manipulating:

$$\begin{aligned}
\frac{d\mathcal{R}}{dp} &= \frac{\partial\mathcal{R}}{\partial p} + \frac{\partial\mathcal{R}}{\partial\psi} \frac{d\psi}{dp} \\
&= \frac{\partial}{\partial p} \left\{ \langle \mathcal{L}\psi \rangle_{E,\mathcal{D},\Omega} - \langle \psi^\dagger, H\psi \rangle_{E,\mathcal{D},\Omega} + \langle \psi^\dagger, G\psi \rangle_{E,\mathcal{D},\Omega} + \langle \psi^\dagger, S_0 \rangle_{E,\mathcal{D},\Omega} \right\} \\
&\quad + \frac{\partial}{\partial\psi} \left\{ \langle \mathcal{L}\psi \rangle_{E,\mathcal{D},\Omega} - \langle \psi^\dagger, H\psi \rangle_{E,\mathcal{D},\Omega} + \langle \psi^\dagger, G\psi \rangle_{E,\mathcal{D},\Omega} \right. \\
&\quad \left. + \langle \psi^\dagger, S_0 \rangle_{E,\mathcal{D},\Omega} \right\} \frac{d\psi}{dp} \\
&= \left\langle \frac{\partial\mathcal{L}\psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial[H-G]\psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \psi^\dagger, \frac{\partial S_0}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \\
&\quad + \frac{\partial}{\partial\psi} \left\{ \langle \mathcal{L}\psi \rangle_{E,\mathcal{D},\Omega} - \langle \psi, H^\dagger\psi^\dagger \rangle_{E,\mathcal{D},\Omega} + \langle \psi, G^\dagger\psi^\dagger \rangle_{E,\mathcal{D},\Omega} \right. \\
&\quad \left. + \langle \psi^\dagger, S_0 \rangle_{E,\mathcal{D},\Omega} \right\} \frac{d\psi}{dp} \\
&= \left\langle \frac{\partial\mathcal{L}\psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial[H-G]\psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \psi^\dagger, \frac{\partial S_0}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \\
&\quad + \left\langle \mathcal{L} - H^\dagger\psi^\dagger + G^\dagger\psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} \frac{d\psi}{dp}.
\end{aligned}$$

We now must consider the computability of each term. The operator Jacobians with respect to p (e.g. $\frac{\partial\mathcal{L}\psi}{\partial p}$) are straightforward, although sometimes cumbersome, to compute using partial differentiation. The total derivative, $\frac{d\psi}{dp}$, on the other hand, is the full Jacobian derivative of the “forward” solution vector ψ with respect to *each* parameter. This is hopeless to know or compute, as it is at least as difficult to compute as $\frac{d\mathcal{R}}{dp}$! This term can be eliminated, however, if the adjoint variable satisfies $H^\dagger\psi^\dagger - G^\dagger\psi^\dagger = \mathcal{L}$, the same adjoint equation as in the previous example. If we can solve this equation for ψ^\dagger , we are left only to evaluate

$$\frac{d\mathcal{R}}{dp} = \left\langle \frac{\partial\mathcal{L}\psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial[H-G]\psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \psi^\dagger, \frac{\partial S_0}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \quad (2.21)$$

to compute a derivative with respect to each entry in p . Again, these terms may be cumbersome to compute, but they are straightforward. It is imperative to note, however, that the expression requires a solution to both the forward problem (ψ), and the adjoint problem, (ψ^\dagger). In addition, it is important to note that one inner product in the equation for $\frac{dR}{dp}$ involves both the forward and adjoint solution. Thus, the cost to compute the full gradient of the QOI with respect to p is one forward solve, one adjoint solve, and one evaluation of Eq. (2.21) per entry in p , and the computation involves products of the forward and adjoint solutions.

Compare this cost to an approach based on finite-difference approximations to $\frac{dR}{dp}$. The most naive example requires a “base solve”, using an unperturbed p vector, and another “perturbed solve” for each entry in p . Other sampling algorithms are available, but all scale linearly (or worse) with the number of entries in p . This property of the adjoint approach, along with other theory to be developed later, is the basis of our claim that this approach is the most effective for producing UQSA information in high-dimensional problems (that is, problems with a long list of p ’s and a high cost for evaluating the governing equations).

2.4.2 The form of the adjoint operators corresponding to the forward transport equation

Using these benefits as motivation, we now derive the form of the continuous adjoint operators, H^\dagger and G^\dagger , given the inner product defined as the integral over all energy, space, and direction. The task will be to manipulate the expressions $\psi^\dagger H \psi$ and $\psi^\dagger G \psi$ to move the adjoint flux to the right side of the operators. We first examine the advection term in H . Writing the inner product in terms of the phase

space integrals,

$$\left\langle \psi^\dagger \Omega \cdot \nabla \psi \right\rangle_{E,\mathcal{D},\Omega} = \int dE \int d\Omega \int dr \psi^\dagger \Omega \cdot \nabla \psi,$$

and using the following result from integration by parts,

$$\int_{\mathcal{D}} dr \psi^\dagger \Omega \cdot \nabla \psi = \int_{\mathcal{D}} dr \nabla \cdot (\psi^\dagger \Omega \psi) - \int_{\mathcal{D}} dr \psi \Omega \cdot \nabla \psi^\dagger$$

and an application of the divergence theorem,

$$\int_{\mathcal{D}} dr \nabla \cdot (\psi^\dagger \Omega \psi) = \oint_{\Gamma} \psi^\dagger \psi (\Omega \cdot \hat{n}) dA$$

we write

$$\left\langle \psi^\dagger \Omega \cdot \nabla \psi \right\rangle_{E,\mathcal{D},\Omega} = - \left\langle \psi \Omega \cdot \nabla \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \oint_{\Gamma} \psi^\dagger \psi (\Omega \cdot \hat{n}) dA \right\rangle_{E,\Omega}.$$

If we expand the boundary term in the previous equation into incident and exiting components

$$\begin{aligned} \left\langle \oint_{\Gamma} \psi^\dagger \psi (\Omega \cdot \hat{n}) dA \right\rangle_{E,\Omega} &= \int dE \int_{\Omega \cdot \hat{n} < 0} d\Omega \oint_{\Gamma} \psi^\dagger \psi (\Omega \cdot \hat{n}) dA \\ &\quad + \int dE \int_{\Omega \cdot \hat{n} > 0} d\Omega \oint_{\Gamma} \psi^\dagger \psi (\Omega \cdot \hat{n}) dA \end{aligned}$$

and assume that the vacuum condition given in problem (2.15) holds, we can eliminate the incident ($\Omega \cdot \hat{n} < 0$) contribution. If we also impose a zero-exiting condition on the adjoint solution, that is

$$\psi^\dagger = 0 \quad \text{on } r \in \Gamma_+$$

the boundary term is eliminated all-together. For the time being, we will maintain the vacuum and zero-exiting conditions on the forward and adjoint solution, respectively. The resulting analysis holds for other boundary conditions, but Eq. (2.17) must be modified to include the boundary term. In summary, we showed that the adjoint to the advection term has a minus sign (this is known as a skew adjoint operator).

The reaction term in H is clearly self-adjoint because $\langle \psi^\dagger, \Sigma\psi \rangle = \langle \psi, \Sigma\psi^\dagger \rangle$.

Moving to the scattering term, we write and manipulate our inner product:

$$\begin{aligned}
& \left\langle \psi^\dagger(r, E, \Omega) \int dE' \int d\Omega' \Sigma_s(r, E' \rightarrow E, \Omega' \cdot \Omega) \psi(r, E', \Omega') \right\rangle_{E, \mathcal{D}, \Omega} \\
&= \int dr \int dE \int d\Omega \psi^\dagger(r, E, \Omega) \int dE' \int d\Omega' \Sigma_s(r, E' \rightarrow E, \Omega' \cdot \Omega) \psi(r, E', \Omega') \\
&= \int dr \int dE \int d\Omega \int dE' \int d\Omega' \psi(r, E', \Omega') \Sigma_s(r, E' \rightarrow E, \Omega' \cdot \Omega) \psi^\dagger(r, E, \Omega) \\
&= \int dr \int dE \int d\Omega \int dE' \int d\Omega' \psi(r, E, \Omega) \Sigma_s(r, E \rightarrow E', \Omega \cdot \Omega') \psi^\dagger(r, E', \Omega') \\
&= \left\langle \psi(r, E, \Omega) \int dE' \int d\Omega' \Sigma_s(r, E \rightarrow E', \Omega \cdot \Omega') \psi^\dagger(r, E', \Omega') \right\rangle_{E, \mathcal{D}, \Omega}.
\end{aligned}$$

A very similar series of manipulations gives the adjoint operator corresponding to the fission multiplication term in G , namely

$$\begin{aligned}
& \left\langle \psi^\dagger(r, E, \Omega) \chi(E) \int dE' \nu \Sigma_f(r, E') \int d\Omega' \psi(r, E', \Omega') \right\rangle_{E, \mathcal{D}, \Omega} \\
&= \left\langle \psi(r, E, \Omega) \nu \Sigma_f(r, E) \int dE' \chi(E') \int d\Omega' \psi^\dagger(r, E', \Omega') \right\rangle_{E, \mathcal{D}, \Omega}.
\end{aligned}$$

We have now defined the form of each term in the adjoint operator $H^\dagger - G^\dagger$ such

that Eqs. (2.17) and (2.18) hold true:

$$\begin{aligned}
H^\dagger \psi^\dagger - G^\dagger \psi^\dagger &= -\Omega \cdot \nabla \psi^\dagger(r, E, \Omega) + \Sigma_t(r, E) \psi^\dagger(r, E, \Omega) \\
&\quad - \int dE' \int d\Omega' \Sigma_s(r, E \rightarrow E', \Omega \cdot \Omega') \psi^\dagger(r, E', \Omega') \\
&\quad - \nu \Sigma_f(r, E) \int dE' \chi(E') \int d\Omega' \psi^\dagger(r, E', \Omega'),
\end{aligned} \tag{2.22}$$

There are a number of possible interpretations of this result. One that is particularly interesting is that the physics appear to occur backwards. For example, the $(-)$ sign in front of the transport term can be interpreted to mean that the neutrons transport in the $(-\Omega)$ direction. Similarly, the scattering and fission source terms contain integrals over *final* energies, as opposed to the initial energies[20], so that if a physical problem has only downscattering (for example), the corresponding adjoint problem has only upscattering.

2.4.3 Modification of the forward solver to solve for the adjoint flux

At first glance, it appears that the inversion of the operator in Eq. (2.22) may require substantial modifications to the transport code used to invert the forward problem. For example, the energy transfer operations must be transposed in order to integrate over final energies. This is not particularly worrisome. The change of sign in front of the transport term is more daunting, however, because it will directly affect the spatial discretization schemes used in the code. In this section, we will show that it is possible to avoid the latter complication by “tricking” the code into solving for $\psi^\dagger(-\Omega)$.

We begin by writing the S_N version of the adjoint operators for direction k :

$$H^\dagger \psi_k^\dagger = -\Omega_k \cdot \nabla \psi^\dagger(r, E, \Omega_k) + \Sigma_t(r, E) \psi^\dagger(r, E, \Omega_k) \\ - [S^\dagger \psi^\dagger(\Omega)](\Omega_k) - [F^\dagger \psi^\dagger(\Omega)](\Omega_k)$$

where $[S^\dagger \psi(\Omega)](\Omega_k)$ and $[F^\dagger \psi(\Omega)](\Omega_k)$ are the adjoint scattering and adjoint fission source contributions to direction k , respectively. We now make a change of variable by letting $\Omega_m = -\Omega_k$ and $\tilde{\psi}^\dagger(\Omega) = \psi^\dagger(-\Omega)$ and insert these changes into our adjoint operator expression,

$$H^\dagger \psi_m^\dagger = \Omega_m \cdot \nabla \tilde{\psi}^\dagger(r, E, \Omega_m) + \Sigma_t(r, E) \tilde{\psi}^\dagger(r, E, \Omega_m) \\ - [S^\dagger \psi^\dagger(\Omega)](-\Omega_m) - [F^\dagger \psi^\dagger(\Omega)](-\Omega_m). \quad (2.23)$$

We now take a closer look at the source terms. We first show that $\int_{4\pi} f(\Omega) d\Omega = \int_{4\pi} f(-\Omega) d\Omega$. The definition of an integral over all angles is

$$\int_{4\pi} f(\Omega) d\Omega = \int_{2\pi} \int_{-1}^1 f(\omega, \mu) d\mu d\omega$$

where μ is the polar cosine and ω is the azimuthal angle in the $x - y$ plane. We now denote $\Omega' = -\Omega$, defined by

$$\mu' = -\mu, \\ \omega' = \begin{cases} \omega + \pi & 0 \leq \omega \leq \pi \\ \omega - \pi & \pi \leq \omega \leq 2\pi \end{cases}$$

which gives $d\mu' = -d\mu$ and $d\omega' = -d\omega$. Now making the change of variable in the

integration,

$$\begin{aligned}
\int_{4\pi} f(\Omega) d\Omega &= \int_{2\pi} \int_{-1}^1 f(\omega, \mu) d\mu d\omega \\
&= \int_{2\pi} \int_1^{-1} -f(\omega', \mu') d\mu' d\omega' \\
&= \int_{2\pi} \int_{-1}^1 f(\omega', \mu') d\mu' d\omega' \\
&= \int_{4\pi} f(\Omega') d\Omega'
\end{aligned}$$

Using this result and the assumption that the fission source is isotropic in angle, we can write

$$[F^\dagger \psi^\dagger(\Omega)](-\Omega_m) = [F^\dagger \tilde{\psi}^\dagger(\Omega)](\Omega_m).$$

We now turn to the scattering source. The typical technique in S_N codes is to write this source in terms of an expansion in the spherical harmonic functions, $Y_\ell^m(\Omega)$. The expansion is

$$[S^\dagger \psi^\dagger(\Omega)](\Omega_m) = \int_0^\infty \sum_{\ell=0}^\infty \sum_{m=-\ell}^{+\ell} \frac{2\ell+1}{4\pi} \sigma_\ell(E \rightarrow E') Y_\ell^m(\Omega_m) \int_{4\pi} \psi^\dagger(\Omega, E) Y_\ell^m(\Omega) d\Omega dE,$$

where σ_ℓ is the ℓ^{th} moment of the scattering cross section. Referring back to Eq. (2.23), we wish to write the contribution of the scattering source to $-\Omega_m$. Making this change, noting that $Y_\ell^m(-\Omega) = (-1)^\ell Y_\ell^m(\Omega)$, and again using the result

$\int_{4\pi} f(\Omega) d\Omega = \int_{4\pi} f(-\Omega) d\Omega$, our representation becomes

$$\begin{aligned}
& \left[S^\dagger \psi^\dagger(\Omega) \right](-\Omega_m) \\
&= \int_0^\infty \sum_{\ell=0}^\infty \sum_{m=-\ell}^{+\ell} \frac{2\ell+1}{4\pi} \sigma_\ell(E \rightarrow E') (-1)^\ell Y_\ell^m(\Omega_m) \\
&\quad \int_{4\pi} \psi^\dagger(\Omega, E) Y_\ell^m(\Omega) d\Omega dE \\
&= \int_0^\infty \sum_{\ell=0}^\infty \sum_{m=-\ell}^{+\ell} \frac{2\ell+1}{4\pi} \sigma_\ell(E \rightarrow E') (-1)^\ell Y_\ell^m(\Omega_m) \\
&\quad \int_{4\pi} \psi^\dagger(-\Omega, E) Y_\ell^m(\Omega) (-1)^\ell d\Omega dE \\
&= \int_0^\infty \sum_{\ell=0}^\infty \sum_{m=-\ell}^{+\ell} \frac{2\ell+1}{4\pi} \sigma_\ell(E \rightarrow E') Y_\ell^m(\Omega_m) \\
&\quad \int_{4\pi} \psi^\dagger(-\Omega, E) Y_\ell^m(\Omega) d\Omega dE \\
&= \left[S^\dagger \tilde{\psi}^\dagger(\Omega) \right](\Omega_m)
\end{aligned}$$

In other words, the scattering source contribution to angle $-\Omega_m$ computed using $\psi(\Omega)$ is equal to the scattering source contribution to angle Ω_m computed using $\psi(-\Omega)$. This result is predicated on the assumption or that the scattering cross section depends only on the cosine of the angle between the incident and exiting directions.

Substituting these results back into Eq. (2.23), we write

$$\begin{aligned}
H^\dagger \psi_m^\dagger &= \Omega_m \cdot \nabla \tilde{\psi}^\dagger(r, E, \Omega_m) + \Sigma_t(r, E) \tilde{\psi}^\dagger(r, E, \Omega_m) \\
&\quad - \left[S^\dagger \tilde{\psi}^\dagger(\Omega) \right](\Omega_m) - \left[F^\dagger \tilde{\psi}^\dagger(\Omega) \right](\Omega_m)
\end{aligned}$$

and note that, with the exception of the transpose of the energy transfer processes, this operator is identical to the forward transport operator! The catch is that the

solution, $\tilde{\psi}_m^\dagger$, is actually the adjoint angular flux in the direction $-\Omega_m$, a distinction which must be kept in mind by the modeler or user of the code. Nonetheless, the same machinery used to invert Eq. (2.14) may be used to invert the adjoint operator, saving time and effort in the development and verification process.

2.4.4 The adjoint Bateman operator

The adjoint Bateman operator is simple and straightforward to derive. We must first recall that the nuclide densities are treated as a list of densities, per nuclide, per cell; that is, we assume a constant density of each element in each spatial cell. Keeping this in mind, we first introduce a new operator b which satisfies

$$\langle bN \rangle_{E,\mathcal{D},\Omega} = BN.$$

This notation will allow us to work with respect to the same inner product as the transport equation. The continuous form of b corresponding to the time-derivative of unknown nuclide density N_i in spatial cell c is

$$\begin{aligned} (b_c(E, \Omega, r)N)_i &= \frac{\sum_j \lambda_{j \rightarrow i} N_{jc} - \lambda_i N_{ic}}{\langle \rangle_{E,\mathcal{D},\Omega}} \\ &+ \frac{\sum_j F_b^{cm} \left(\sigma_{j \rightarrow i}(E) \psi(E, \Omega, r) N_{jc} - \delta_{ij} \sigma_{a,i}(E) \psi(E, \Omega, r) N_{ic} \right)}{\langle \rangle_{\mathcal{D}_c}} \end{aligned} \quad (2.24)$$

where $F_b^{cm}=1.0e-24$ is the conversion from cm^2 to b , and \mathcal{D}_c is the domain of cell c . If we apply $\langle \rangle_{E,\mathcal{D},\Omega}$ to each term in this equation, we arrive at the familiar Bateman

terms, which are angle and energy integrated and cell averaged:

$$\begin{aligned}
& \int_0^\infty dE \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \left\{ \frac{\sum_j \lambda_{j \rightarrow i} N_{jc} - \lambda_i N_{ic}}{\langle \rangle_{E, \mathcal{D}, \Omega}} \right\} = \sum_j \lambda_{j \rightarrow i} N_{jc} - \lambda_i N_{ic} \\
& \int_0^\infty dE \int_{4\pi} d\Omega \int_{\mathcal{D}} dV \left\{ \frac{\sum_j F_b^{cm} \left(\sigma_{j \rightarrow i}(E) \psi(E, \Omega, r) N_{jc} - \delta_{ij} \sigma_{a,i}(E) \psi(E, \Omega, vr) N_{ic} \right)}{\langle \rangle_{\mathcal{D}_c}} \right\} \\
&= \sum_j F_b^{cm} \int_0^\infty dE \left(\frac{\sigma_{j \rightarrow i}(E) \int_{\mathcal{D}} dV N_{jc} \int_{4\pi} d\Omega \psi(E, \Omega, r)}{\langle \rangle_{\mathcal{D}_c}} \right) \\
&\quad - \int_0^\infty dE \left(\frac{\sigma_{a,i}(E) \int_{\mathcal{D}} dV N_{ic} \int_{4\pi} d\Omega \psi(E, \Omega, r)}{\langle \rangle_{\mathcal{D}_c}} \right) \\
&= \sum_j F_b^{cm} N_{jc} \int_0^\infty dE \left(\frac{\sigma_{j \rightarrow i}(E) \int_{\mathcal{D}_c} dV \int_{4\pi} d\Omega \psi(E, \Omega, r)}{\langle \rangle_{\mathcal{D}_c}} \right) \\
&\quad - N_{ic} \int_0^\infty dE \left(\frac{\sigma_{a,i}(E) \int_{\mathcal{D}_c} dV \int_{4\pi} d\Omega \psi(E, \Omega, r)}{\langle \rangle_{\mathcal{D}_c}} \right) \\
&= \sum_j F_b^{cm} N_{jc} \int_0^\infty dE \sigma_{j \rightarrow i}(E) \int_{4\pi} d\Omega \psi_c(E, \Omega) \\
&\quad - N_{ic} \int_0^\infty dE \sigma_{a,i}(E) \int_{4\pi} d\Omega \psi_c(E, \Omega)
\end{aligned}$$

where ψ_c is a cell-averaged angular flux. The discrete version of Eq. (2.24) is

$$\begin{aligned}
(b_{cgq} N)_i &= \frac{\sum_j \lambda_{j \rightarrow i} N_{jc} - \lambda_i N_{ic}}{\langle \rangle_{E, \mathcal{D}, \Omega}} + \frac{\sum_j F_b^{cm} \left(\sigma_{g,j \rightarrow i} \psi_{egq} N_{jc} - \delta_{ij} \sigma_{a,gi} \psi_{egq} N_{ic} \right)}{\langle \rangle_{\mathcal{D}_c}}
\end{aligned} \tag{2.25}$$

where ψ_{egq} is the angular flux defined on the cell elements.

Because each term in $BN = \langle bN \rangle_{E,\mathcal{D},\Omega}$ is angle and energy integrated and spatially averaged over each cell, BN has no dependence in these variables. Therefore

$$\langle N^\dagger BN \rangle_{E,\mathcal{D},\Omega} = \langle NB^\dagger N^\dagger \rangle_{E,\mathcal{D},\Omega} = \langle NB^T N^\dagger \rangle_{E,\mathcal{D},\Omega},$$

or, the adjoint operator of B with respect to our phase-space integration inner product is just the transpose of B . In later sections, we will also find the following relationship from integration by parts to be useful:

$$\begin{aligned} & \int_{t_0}^{t_f} \left\{ N^\dagger \frac{dN}{dt} - \langle N^\dagger BN \rangle_{E,\mathcal{D},\Omega} \right\} dt \\ &= \left[N^\dagger N \right]_{t=t_0}^{t_f} - \int_{t_0}^{t_f} \left\{ N \frac{dN^\dagger}{dt} + \langle N^\dagger BN \rangle_{E,\mathcal{D},\Omega} \right\} dt \end{aligned} \quad (2.26)$$

2.5 Adjoint Consistency of Spatial and Temporal Discretizations

An important distinction may exist between the continuous adjoint of a particular linear operator and the adjoint derived by first applying a discretization to the linear operator and then taking the adjoint of the resulting set of discrete equations. The former is called the continuous adjoint problem, and the latter is referred to as the discrete adjoint problem. For obvious reasons, a combination of the operator and the discretization scheme determine whether the continuous and discrete adjoint problems are one in the same. If they are the same, the discretization is said to be an *adjoint-consistent* discretization of the operator.

Adjoint-consistent discretizations inherit the smoothness properties of the continuous adjoint solution and have important implications for convergence rates. For example, an adjoint-consistent linear discontinuous spatial discretization will approach the continuous adjoint solution at $\mathcal{O}(h^3)$ while an inconsistent discretization will converge at a suboptimal $\mathcal{O}(h^2)$, where h is the characteristic mesh size. In the following subsections, we show that the discontinuous Galerkin (DG) spatial discretization scheme is adjoint-consistent, and we review the conditions necessary for adjoint consistency in a Runge-Kutta scheme.

2.5.1 Discontinuous Galerkin spatial discretization

In this subsection, we review a publication by Hartmann[21], which shows that the linear DG method applied to the neutron transport equation is indeed an adjoint consistent discretization. Consider the transport problem written as

$$\begin{cases} \nabla \cdot \Omega \psi + \sigma_t \psi = q_{tot}(r, E, \Omega) + S_0 & r \in \mathcal{D} \\ \psi = \psi_B & \text{on } r \in \Gamma_- \end{cases}$$

where again $\Gamma_- = \{r \in \partial\mathcal{D} \mid \Omega \cdot \hat{n} < 0\}$, the scattering and fission sources have been combined into a single source term (a representation consistent with a typical sweep algorithm), and we have temporarily allowed for non-vacuum Dirichlet boundary conditions. Now let \mathcal{D} be subdivided into shape-regular mesh elements $\kappa \in \tau_h$ and let $V_h \subset V$ be the discrete space of linear functions on τ_h . The linear DG discretization can then be written in terms of a bilinear operator $\mathcal{B} : V \times V \rightarrow \mathbb{R}$ as follows: Find $\psi_h \in V_h$ such that

$$\mathcal{B}(\psi_h, v) = L(v) \quad \forall v \in V \quad (2.27)$$

where $L : V \rightarrow \mathbb{R}$ is a linear operator containing the source term and possibly boundary forcing terms. Now consider the task of computing a response, written previously as

$$\mathcal{R}(\psi) = \left\langle \int_{\mathcal{D}} \mathcal{L}\psi \right\rangle_{E,\Omega}.$$

We have shown that this QOI leads to the adjoint problem

$$\begin{cases} H^\dagger \psi^\dagger = \mathcal{L} & r \in \mathcal{D} \\ \psi^\dagger = 0 & \text{on } r \in \Gamma_+ \end{cases}$$

The discretization Eq. (2.27) is said to be adjoint consistent if the continuous (exact) adjoint, $\psi^\dagger \in V$ satisfies

$$\mathcal{B}(w, \psi^\dagger) = \mathcal{R}(w) \quad \forall w \in V. \quad (2.28)$$

In what follows, we give the bilinear form of the DG discretization applied to the

forward problem and show that this discretization is indeed adjoint consistent. We adopt the following notation: let κ^i and κ^j be adjacent elements in τ_h sharing edge $\partial\kappa$ with unit normal \hat{n}^{ij} pointing from κ^i to κ^j . Then, for the scalar valued function g , let g^k denote the value of g on $\partial\kappa$ taken from the inside of element k , $k = i, j$.

The form of $\mathcal{B}(\cdot, \cdot)$ is found by multiplying the transport operator by the test function v , integrating over \mathcal{D} , and applying integration by parts:

$$\begin{aligned}\mathcal{B}(\psi_h, v) &= \int_{\mathcal{D}} \nabla \cdot (\Omega \psi_h) v + \int_{\mathcal{D}} \sigma_t \psi_h v \\ &= \int_{\mathcal{D}} -\Omega \psi_h \cdot \nabla v + \int_{\mathcal{D}} \sigma_t \psi_h v + \sum_{\kappa^i \in \tau_h} \int_{\partial\kappa_- \setminus \Gamma} (\Omega \cdot \hat{n}^{ij}) \psi_h^j v^i \\ &\quad + \sum_{\kappa^i \in \tau_h} \int_{\partial\kappa_+} (\Omega \cdot \hat{n}^{ij}) \psi_h^i v^i.\end{aligned}\tag{2.29}$$

Note that we have applied the upwinding rule to the scheme on the inflow boundaries of each element, and we have moved a boundary forcing term to the right-hand-side such that

$$L(v) = \int_{\mathcal{D}} q_{tot} v + \int_{\mathcal{D}} S_0 v - \int_{\Gamma_-} (\Omega \cdot \hat{n}) \psi_B v.$$

The first term in Eq. (2.29) may be broken into a sum of integrals over the elements, as follows

$$\int_{\mathcal{D}} -\Omega \psi_h \cdot \nabla v = \sum_{\kappa^i \in \tau_h} \int_{\kappa^i} -\Omega \psi_h \cdot \nabla v.$$

If we apply integration by parts to each of these local integrals (and do not apply

upwinding), we have

$$\begin{aligned}
\int_{\kappa^i} -\Omega \psi_h^i \cdot \nabla v &= \int_{\kappa^i} \nabla \cdot (\Omega \psi_h) v - \int_{\kappa^i} \nabla \cdot (\Omega \psi_h v) \\
&= \int_{\kappa^i} \nabla \cdot (\Omega \psi_h) v - \int_{\partial \kappa_-} (\Omega \cdot \hat{n}^{ij}) \psi_h^i v^i - \int_{\partial \kappa_+} (\Omega \cdot \hat{n}^{ij}) \psi_h^i v^i \\
&= \int_{\kappa^i} \nabla \cdot (\Omega \psi_h) v - \int_{\partial \kappa_- \setminus \Gamma} (\Omega \cdot \hat{n}^{ij}) \psi_h^i v^i \\
&\quad - \int_{\Gamma_-} (\Omega \cdot \hat{n}) \psi_h v - \int_{\partial \kappa_+} (\Omega \cdot \hat{n}^{ij}) \psi_h^i v^i.
\end{aligned}$$

Substituting this result back into Eq. (2.29) we obtain

$$\begin{aligned}
\mathcal{B}(\psi_h, v) &= \int_{\mathcal{D}} \nabla \cdot (\Omega \psi_h) v + \int_{\mathcal{D}} \sigma_t \psi_h v - \sum_{\kappa^i \in \tau_h} \int_{\partial \kappa_- \setminus \Gamma} (\Omega \cdot \hat{n}^{ij}) [[\psi]] v^i \\
&\quad - \int_{\Gamma_-} (\Omega \cdot \hat{n}) \psi_h v,
\end{aligned} \tag{2.30}$$

where the notation $[[g]]$ is used to denote the jump in the function g across $\partial \kappa$, $g^i - g^j$.

It is instructive at this point to show that this discretization is consistent with the continuous transport equation. Rewrite the system $\mathcal{B}(\psi_h, v) = L(v)$ by combining terms with like-integrals as

$$\int_{\mathcal{D}} R_{\mathcal{D}}(\psi_h) v + \sum_{\kappa^i \in \tau_h} \int_{\partial \kappa_- \setminus \Gamma} R_-(\psi_h) v + \int_{\Gamma_-} R_B(\psi_h) v = 0,$$

where $R_{\mathcal{D}}$, R_- , and R_B are interior, interior edge, and boundary residuals, respectively. For our transport equation, we have

$$\begin{aligned}
R_{\mathcal{D}} &= q_{tot} + S_0 - \left(\nabla \cdot (\Omega \psi_h) + \sigma_t \psi_h \right) & r \in \mathcal{D} \\
R_- &= (\Omega \cdot \hat{n}^{ij}) [[\psi_h]] & r \in \partial \kappa \\
R_B &= (\Omega \cdot \hat{n} \left(\psi_h - \psi_B \right)) & r \in \Gamma_-
\end{aligned}$$

Each residual evaluated at $\psi_h = \psi$ vanishes. In other words, the linear DG discretization is said to be consistent because Eq. (2.27) evaluated at the continuous solution is satisfied, that is

$$\mathcal{B}(\psi, v) = L(v) \quad \forall v \in V.$$

We can now move towards the proof of adjoint consistency for the DG discretization of the transport equation. We re-write the adjoint operator into sweep-algorithm form

$$-\nabla \cdot (\Omega \psi^\dagger) + \sigma_t \psi^\dagger = \mathcal{L} + \tilde{q}_{tot}^\dagger,$$

with boundary condition $\psi^\dagger(r = \Gamma_+) = 0$. Note that we have gathered the adjoint scattering and adjoint fission operators into a shorthand \tilde{q}_{tot}^\dagger . The bilinear form of the left hand side of this equation is equivalent to the first two terms in Eq. (2.29). To incorporate the adjoint boundary conditions, we make the substitution

$$\sum_{\kappa^i \in \tau_h} \int_{\partial\kappa_- \setminus \Gamma} (\Omega \cdot \hat{n}^{ij}) \psi_h^j v^i = - \sum_{\kappa^i \in \tau_h} \int_{\partial\kappa_+ \setminus \Gamma} (\Omega \cdot \hat{n}^{ij}) \psi_h^i v^j,$$

to write

$$\begin{aligned}
\mathcal{B}(\psi, v_h) &= \int_{\mathcal{D}} -\Omega \psi \cdot \nabla v_h + \int_{\mathcal{D}} \sigma_t \psi v_h - \sum_{\kappa^i \in \tau_h} \int_{\partial \kappa_+ \setminus \Gamma} (\Omega \cdot \hat{n}^{ij}) \psi^i v_h^j \\
&\quad + \sum_{\kappa^i \in \tau_h} \int_{\partial \kappa_+} (\Omega \cdot \hat{n}^{ij}) \psi^i v_h^i \\
&= \int_{\mathcal{D}} -\Omega \psi \cdot \nabla v_h + \int_{\mathcal{D}} \sigma_t \psi v_h + \sum_{\kappa^i \in \tau_h} \int_{\partial \kappa_+ \setminus \Gamma} (\Omega \cdot \hat{n}^{ij}) \psi [[v_h]] \\
&\quad + \int_{\Gamma_+} (\Omega \cdot \hat{n}) \psi v_h.
\end{aligned} \tag{2.31}$$

Forming our interior, interior edge, and boundary residuals again, we find

$$\begin{aligned}
R_{\mathcal{D}}^\dagger(v_h) &= \mathcal{L} + \tilde{q}_{tot}^\dagger + \Omega \cdot \nabla v_h - \sigma_t v_h \quad r \in \mathcal{D} \\
R_- &= (\Omega \cdot \hat{n}^{ij}) [[v_h]] \quad r \in \partial \kappa \\
R_B &= (\Omega \cdot \hat{n}) v_h \quad r \in \Gamma_+.
\end{aligned}$$

Again, each of these residuals vanishes when evaluated at the continuous (exact) adjoint solution, $v_h = \psi^\dagger$. Therefore, the continuous adjoint satisfies the bilinear form

$$\mathcal{B}(w, \psi^\dagger) = \mathcal{L}(w) \quad \forall w \in V$$

and the DG method for the neutron transport equation is adjoint-consistent.

2.5.2 Runge-Kutta time discretization

When we derived Eq. (2.26), we leveraged the continuous relationship

$$\int_{t_0}^{t_f} N^\dagger \frac{dN}{dt} dt = \left[N^\dagger N \right]_{t_0}^{t_f} - \int_{t_0}^{t_f} \frac{dN^\dagger}{dt} N dt$$

to develop the adjoint Bateman equation. It is yet to be shown, however, that this relationship holds after discretization in time. Specifically, in order for the discrete versions of this relationship to hold, the scheme must preserve the corresponding relationship over each time step $t \in [t_k, t_{k+1}]$

$$\int_{t_k}^{t_{k+1}} N^\dagger \frac{dN}{dt} dt = \left[N^\dagger N \right]_{t_k}^{t_{k+1}} - \int_{t_k}^{t_{k+1}} \frac{dN^\dagger}{dt} N dt.$$

Then a simple summation over k will telescope into the continuous relationship. In this section, we derive the conditions under which a Runge-Kutta time discretization scheme (covered in Sec. 2.3) will preserve this property.

To introduce some notation, we begin with a quick re-derivation of the continuous adjoint equations for a general non-linear system of equations. Consider the task of solving the system

$$\begin{aligned} x' &= f(t, x), \quad t \in [t_0, t_f] \\ x(t_0) &= x_0 \end{aligned} \tag{2.32}$$

and computing the sensitivity of some quantity of interest, $Q \equiv Q(x(t_f))$, a functional of the terminal solution, to a vector of parameters p . To derive an adjoint equation, we introduce a multiplier x^\dagger for the state equations, integrate the product over time, subtract the result from the QOI to form a Lagrangian, and apply integration by parts:

$$\begin{aligned} \mathcal{L} &= Q - \int_{t_0}^{t_f} x^\dagger \left(x' - f(t, x) \right) dt \\ &= Q - \left[x^\dagger x \right]_{t_0}^{t_f} + \int_{t_0}^{t_f} (x^\dagger)' x dt + \int_{t_0}^{t_f} x^\dagger f(t, x) dt \end{aligned}$$

Now the derivative of the Lagrangian is

$$\begin{aligned} \frac{d}{dp}\mathcal{L} = \frac{dQ}{dp} &= \left[\nabla_x Q \frac{dx}{dp} + \nabla_p Q - x^\dagger \frac{dx}{dp} \right]_{t_f} + x^\dagger \frac{dx}{dp} \Big|_{t_0} \\ &\quad + \int_{t_0}^{t_f} \left[(x^\dagger)' + x^\dagger \nabla_x f(t, x) \right] \frac{dx}{dp} dt + \int_{t_0}^{t_f} x^\dagger \nabla_p f(t, x) dt, \end{aligned}$$

where ∇_x and ∇_p indicate partial derivatives. Now, if x^\dagger satisfies the following system of equations,

$$\begin{aligned} (x^\dagger)' &= -x^\dagger \nabla_x f(t, x), \quad t \in [t_0, t_f] \\ x^\dagger(t_f) &= \nabla_x Q \end{aligned} \tag{2.33}$$

then we have an adjoint-based expression for the full gradient (sensitivity) of Q w.r.t. the parameter vector:

$$\frac{dQ}{dp} = \nabla_p Q + x^\dagger \frac{dx}{dp} \Big|_{t_0} + \int_{t_0}^{t_f} x^\dagger \nabla_p f(t, x) dt.$$

We now move to an analysis of the discretized version of system 2.32 using a general s -stage Runge-Kutta method. The analysis follows the work of Hager[22], who was interested in similar properties for optimal control problems. The discrete system in time step $t \in [t_k, t_{k+1}]$ is

$$x'_k = \sum_{i=1}^s b_i f\left(t_i, y_i\right) \tag{2.34}$$

$$y_i = x_k + h_k \sum_{j=1}^s a_{ij} f\left(t_j, y_j\right) \tag{2.35}$$

$$x(t_0) = x_0$$

where the y_i are stage vectors, h_k is the time step (may be non-uniform), and x'_k is

the usual forward difference expression, $x'_k = \frac{x_{k+1} - x_k}{h_k}$. We note that the stage vectors are different in each time step, but we do not explicitly write this dependence.

We will assume that the discrete integration by parts rule holds, *i.e.*

$$\sum_{k=1}^N x_k^\dagger x'_k = \left[x^\dagger x \right]_{t_1}^{t_{N+1}} - \sum_{k=1}^N (x_k^\dagger)' x_k,$$

and proceed by introducing multiplier x^\dagger for Eq. 2.34 and multiplier y^\dagger for a rearrangement of Eq. (2.35) written as

$$0 = \frac{y_j - x_k}{h_k} - \sum_{i=1}^s a_{ji} f(t_i, y_i) \equiv y'_j - \sum_{i=1}^s a_{ji} f(t_i, y_i)$$

The discrete Lagrangian is

$$\begin{aligned} \mathcal{L} &= Q - \sum_{k=1}^N \left\{ x_k^\dagger \left[x'_k - \sum_{i=1}^s b_i f(t_i, y_i) \right] + \sum_{j=1}^s y_j^\dagger \left[y'_j - \sum_{i=1}^s a_{ji} f(t_i, y_i) \right] \right\} \\ &= Q - \left[x^\dagger x \right]_{t_1}^{t_{N+1}} + \sum_{k=1}^N \left\{ (x_k^\dagger)' x_k + x_{k+1}^\dagger \sum_{i=1}^s b_i f(t_i, y_i) \right. \\ &\quad \left. + \sum_{j=1}^s y_j^\dagger \sum_{i=1}^s a_{ji} f(t_i, y_i) - \sum_{j=1}^s y_j^\dagger y'_j \right\} \end{aligned}$$

We now take the full derivative by applying the following operator to \mathcal{L} :

$$\frac{d\mathcal{L}}{dp} = \nabla_x \mathcal{L} \frac{dx}{dp} + \nabla_y \mathcal{L} \frac{dy}{dp} + \nabla_p \mathcal{L}.$$

The terms are

$$\begin{aligned}\nabla_x \mathcal{L} \frac{dx}{dp} &= \left[\left[\nabla_x Q - x^\dagger \right] \frac{dx}{dp} \right]_{t_N} + \sum_{k=1}^N \left\{ \left[(x_k^\dagger)' + \sum_{j=1}^s \frac{y_j^\dagger}{h_k} \right] \frac{dx}{dt} \right\} \\ \nabla_y \mathcal{L} \frac{dy}{dp} &= \sum_{k=1}^N \left\{ \left[x_{k+1}^\dagger \sum_{i=1}^s b_i \nabla_y f(t_i, y_i) - \sum_{j=1}^s \frac{y_j^\dagger}{h_k} \right. \right. \\ &\quad \left. \left. + \sum_{j=1}^s y_j^\dagger \sum_{i=1}^s a_{ji} \nabla_y f(t_i, y_i) \right] \frac{dy}{dp} \right\} \\ \nabla_p \mathcal{L} &= \sum_{k=1}^N \left\{ x_{k+1}^\dagger \sum_{i=1}^s b_i \nabla_p f(t_i, y_i) + \sum_{j=1}^s y_j^\dagger \sum_{i=1}^s a_{ji} \nabla_p f(t_i, y_i) \right\}\end{aligned}$$

From these terms we extract the following adjoint equations:

$$(x_k^\dagger)' = - \sum_{j=1}^s \frac{y_j^\dagger}{h_k} \quad \text{or} \quad x_k^\dagger - x_{k+1}^\dagger = \sum_{j=1}^s y_j^\dagger, \quad (2.36)$$

$$x_{t_N}^\dagger = \nabla_x Q(t_N), \quad (2.37)$$

$$y_i^\dagger = h_k \left[x_{k+1}^\dagger b_i + \sum_{j=1}^s y_j^\dagger a_{ji} \right] \nabla_y f(t_i, y_i). \quad (2.38)$$

We now require the (fairly weak) assumption that $b_j > 0$, $j = 1 \dots s$, which is true for most Runge-Kutta schemes. Given this, define a variable

$$\Gamma_i \equiv x_{k+1}^\dagger + \sum_{j=1}^s \frac{a_{ji}}{b_i} y_j^\dagger$$

and rewrite Eq. (2.38) as

$$y_i^\dagger = h_k b_i \Gamma_i \nabla_y f(t_i, y_i). \quad (2.39)$$

If we sum Eq. (2.39) over i and substitute from Eq. (2.36), we find

$$x_k^\dagger = x_{k+1}^\dagger + h_k \sum_i b_i \Gamma_i \nabla_y f(t_i, y_i).$$

Similarly, if we multiply Eq. (2.39) by $\frac{a_{ij}}{b_j}$, sum over i , and substitute our expression for Γ , we write

$$\sum_{i=1}^s \frac{a_{ij} y_i^\dagger}{b_j} = \Gamma_j - x_{k+1}^\dagger = h_k \sum_{i=1}^s \frac{a_{ij}}{b_j} b_i \Gamma_i \nabla_y f(t_i, y_i)$$

After some re-arrangement of these equations, we find an effective Runge-Kutta scheme for the adjoint problem, Eq. (2.33)

$$x_k^\dagger = x_{k+1}^\dagger + h_k \sum_i b_i \Gamma_i \nabla_y f(t_i, y_i) \quad (2.40)$$

$$\Gamma_i = x_{k+1}^\dagger + h_k \sum_{j=1}^s \frac{a_{ji}}{b_i} b_j \Gamma_j \nabla_y f(t_j, y_j) \quad (2.41)$$

$$x_{t_N}^\dagger = \nabla_x Q(t_N) \quad (2.42)$$

In other words, if the following Butcher tableau is used to define the forward Runge-Kutta method

$$\begin{array}{c|c} c & a \\ \hline & b \end{array},$$

then the true adjoint Runge-Kutta method is defined by

$$\begin{array}{c|c} c & \bar{a} \\ \hline & \bar{b} \end{array},$$

where

$$\begin{aligned}\bar{b} &= b \\ \bar{a}_{ij} &= \frac{a_{ji}}{b_i} b_j.\end{aligned}\tag{2.43}$$

Further studies in the field of optimal control show that the adjoint Runge-Kutta scheme maintains the same order of accuracy and linear stability properties of the forward scheme[23, 24].

These authors also point out that some schemes are symmetric in the sense that the forward and adjoint coefficients are the same. For example, consider solving the forward and adjoint problems

$$\left\{ \begin{array}{l} \frac{dx}{dt} = f(t, x) \\ x(t_0) = x_0 \end{array} \right., \quad \left\{ \begin{array}{l} \frac{dx^\dagger}{dt} = -x^\dagger \nabla_x f(t, x) \equiv -H(t, x, x^\dagger) \\ x^\dagger(t_f) = x_f^\dagger \end{array} \right.$$

using the RK4 scheme (see Eq. (E.3)). Using rules (2.43), we find

$$b = \begin{bmatrix} \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{bmatrix}, \quad \bar{a}_{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\bar{b} = \begin{bmatrix} \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{bmatrix}, \quad \bar{a}_{ij} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

If we explicitly write the expression to advance the forward solution over time step

k , we find

$$x_{k+1} = x_k + h_k \left[\frac{1}{6}f(t, x_1) + \frac{1}{3}f(t, x_2) + \frac{1}{3}f(t, x_3) + \frac{1}{6}f(t, x_4) \right]$$

where

$$\begin{aligned} x_1 &= x_k \\ x_2 &= x_k + \frac{1}{2}h_k f(t, x_1) \\ x_3 &= x_k + \frac{1}{2}h_k f(t, x_2) \\ x_4 &= x_k + h_k f(t, x_3) \end{aligned}$$

Similarly, the expression to advance the adjoint solution backward through time step k is

$$x_k^\dagger = x_{k+1}^\dagger + h_k \left[\frac{1}{6}H(t, x, x_1^\dagger) + \frac{1}{3}H(t, x, x_2^\dagger) + \frac{1}{3}H(t, x, x_3^\dagger) + \frac{1}{6}H(t, x, x_4^\dagger) \right]$$

where

$$\begin{aligned} x_1^\dagger &= x_{k+1}^\dagger + h_k H(t, x, x_2^\dagger) \\ x_2^\dagger &= x_{k+1}^\dagger + \frac{1}{2}h_k H(t, x, x_3^\dagger) \\ x_3^\dagger &= x_{k+1}^\dagger + \frac{1}{2}h_k H(t, x, x_4^\dagger) \\ x_4^\dagger &= x_{k+1}^\dagger \end{aligned}$$

The symmetry is in that the progression across the time-step uses the same linear combination of stage vectors and derivatives in each case. In other words, if the

modeler applied the forward Runge-Kutta rule (b and a_{ij}) to this modified adjoint problem for variable \tilde{x}^\dagger

$$\begin{cases} \frac{d\tilde{x}^\dagger}{dt} = H(t, x, \tilde{x}^\dagger) \\ \tilde{x}^\dagger(t_0) = x_f^\dagger \end{cases},$$

the following would be true: $x^\dagger(t_0) = \tilde{x}^\dagger(t_f)$. The explicit-Euler (Eq. (E.1)) and modified-Euler (Eq. (E.2)) schemes also satisfy this symmetry property.

2.6 Review of the Literature Leading to the Current State of the Art in Depletion Perturbation Theory

In Sec. 2.4.1, we derived a sensitivity equation, Eq. (2.21), which provides the gradient of a reactor QOI with respect to a list of parameters at the cost of a single forward and single adjoint transport solve. This is a form of perturbation theory, and the nuclear engineering literature is rich with its development and application to light water reactor analysis. The earliest cited work is often a technical report from the Manhattan project[25]; perturbation theory in its current form, however, began to appear in conference proceedings papers and nuclear engineering textbooks around two decades later[26, 27, 28, 29]. Later, authors worked to formalize the adjoint technique and analyze the mathematical properties of sensitivity theory[30, 31, 32, 33]. Work by these and many other authors showed that the formulation of an adjoint transport equation yields valuable expression for reactor quantities of interest as well as first-order-accurate expressions for the sensitivity of a QOI with respect to cross sections or other parameters.

A number of reactor design codes, such as SCALE[34, 35] and CASMO-5[36], encapsulate the formalism for computing both the QOI itself and its sensitivity with respect to design parameters. Recent work has extended the sensitivity estimates to higher-order accuracy [37] and has generalized the adjoint equations to include feedback from other physics[38] using simple component-like coupling models.

Depletion perturbation theory extends the classic perturbation theory to include the dynamics of the Bateman equations. Pioneering work by Williams[39] and Takeda/Umano[40] developed the equations that constitute a coupled forward and adjoint depletion problem to produce sensitivity estimates for a select set of parameters. We will derive and discuss the adjoint depletion equations corresponding

to both the source-driven and k-eigenvalue forward problems in full detail in later sections. For now, we note that the adjoint equations form a DAE system and can be written in a general way as

$$\begin{aligned}\frac{dN^\dagger}{dt} &= -B^\dagger N^\dagger + S_{N^\dagger}(N, \psi, p, t) \\ H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger &= S_{\psi^\dagger}(N, \psi^\dagger, p, t).\end{aligned}$$

Upon solving these equations, we are left with a sensitivity equation in the form of

$$\frac{d\text{QOI}}{dp} = \int_{t_0}^{t_f} \left\langle \frac{\partial[\text{QOI}]}{\partial p} + N^\dagger \frac{\partial[BN]}{\partial p} + \psi^\dagger \frac{\partial[H\psi - G\psi]}{\partial p} \right\rangle dt. \quad (2.44)$$

The adjoint equations are very similar in structure to the forward equations. A major challenge in solving them, however, is that the evaluation of the adjoint sources (S_{N^\dagger} and S_{ψ^\dagger}) and the sensitivity equation require access to the solution of the forward equations at each time step. Second, in the k-eigenvalue case the adjoint operator ($H^\dagger - \lambda G^\dagger$) is singular, which necessitates special techniques for the adjoint solution. We will discuss the implications of these challenges and our algorithmic approaches in the theory and implementation sections of this dissertation.

Although some of the early theory was developed for the general transport equation, most of the computational examples to date use the simpler diffusion approximation with a constant amplitude function across the depletion cycles. The nuclear power industry successfully uses diffusion-based depletion perturbation theory to develop specialized equations for fuel cycle and shuffling optimization. [41, 42, 43, 44, 45]. In 1988, Yang and Downar [46] developed the adjoint equations corresponding to the burnup equations with the linear, but still diffusion-based, flux amplitude approximation.

Most reported present-day work still uses a diffusion approximation, however, and does not discuss the algorithmic challenges of managing the demand for the forward solution during the adjoint solve, a challenge that will only grow if the full angular-dependent transport solution is required. Moreover, the authors that developed the depletion perturbation theory do not mention its application for estimating global error by propagating local truncation residuals; theory for doing so in other settings is available from the fields of optimization and control[19, 47].

3. THEORY

In this section we present the mathematical foundations of our approach to depletion perturbation theory. Some of the work is extended or borrowed from other fields and is applied to our nuclear engineering problem while other parts are, to the best of our knowledge, original theory and work.

We begin with a detailed derivation of an adjoint framework corresponding to the DAE system (2.6). We show that the adjoint variable can be used to propagate local sensitivities and error estimates to total derivatives and global error estimates, respectively. Borrowing from the fields of optimization and control, we maintain as much generality in this derivation as possible. The result is a flexible framework that facilitates the adjoint approach in a multi-physics modeling environment.

We then apply our framework to derive the specific adjoint equations corresponding to the source-driven and k-eigenvalue depletion equations. We discuss our strategy for integrating the adjoint and sensitivity equations, and we finish with examples of the adjoint equations for three specific QOIs that appear in this dissertation as well as for QOIs that do not immediately conform to our framework.

3.1 A Variational Derivation of the Adjoint Equations Corresponding to Parameter-Dependent DAE Systems

In this section, we derive the adjoint equations corresponding to the general, parameter-dependent DAE system and the resulting adjoint-based expressions for QOI sensitivity and error estimates. We recently published an article[48] containing a similar analysis, which borrows from the work of Cao, Li, and Petzold[19, 49]. In this report, we assumed a semi-discrete DAE system (that is, discretized in every variable except time) and that the inner product defining the adjoint operators was a vector dot product. This is a common assumption in the literature because it provides linearized adjoint equations for the general non-linear system.

For this dissertation, however, our governing equations have a bi-linear property which allows us to define the adjoint operators in terms of an integral inner product. A central contribution of this work is a general approach for producing adjoint equations in a multi-physics modeling environment. Thus, in this section, we maintain a general inner-product notation, $\langle f, g \rangle$, to facilitate the modeler's choice of available inner products for the system of interest. Also, unless otherwise noted, the derivations that follow hold for continuous or discrete systems so long as the inner product is appropriate.

For convenience, we re-produce system (2.6) in index-1 form and include the explicit dependence on the parameter vector p :

$$F(\dot{x}, x, p, t) = \begin{Bmatrix} F^d(\dot{x}, x, p, t) \\ F^a(\dot{x}, x, p, t) \end{Bmatrix} = \begin{Bmatrix} \dot{x}^d - f^d(x, p, t) \\ f^a(x, p, t) \end{Bmatrix} = 0 \quad (3.1)$$

$$x^d(t_0) = x_0,$$

$$t \in [t_0, t_f].$$

Let us introduce notation that will carry through this section. Subscripts on operators indicate partial derivatives, i.e.:

$$F_p \equiv \frac{\partial F}{\partial p}.$$

Subscripts on variables indicate full, functional derivatives, i.e.:

$$x_p \equiv \frac{dx}{dp}.$$

For operators, we will denote functional derivatives explicitly. For example, the i^{th} component of $\frac{dF}{dp}$ is defined as

$$\frac{dF}{dp} \equiv \frac{\partial F}{\partial p} + \frac{\partial F}{\partial N} N_p + \frac{\partial F}{\partial \psi} \psi_p + \dots = \lim_{\epsilon \rightarrow 0} \left\{ \frac{F(p + \epsilon e_i) - F(p)}{\epsilon} \right\},$$

where e_i is a vector of zeros except for a 1 in the i^{th} component. Note that our form of Eq. (3.1) has the property

$$F_{\dot{x}} = \begin{bmatrix} F_{\dot{x}^d}^d & F_{\dot{x}^a}^d \\ F_{\dot{x}^d}^a & F_{\dot{x}^a}^a \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}. \quad (3.2)$$

Although it is not the most general formulation, this property holds for many engineering systems and certainly for the depletion equations considered here.

Recall that our PS&E problem is to solve system (3.1) for $x(t)$ and compute some derived quantity of interest, Q . For now, we will assume that Q can be written as a time-integrated functional of the depletion solution:

$$Q = \int_{t_0}^{t_f} \langle R(x, p, t) \rangle dt. \quad (3.3)$$

For much of this dissertation, we will investigate QOIs that depend only on the depletion solution at $t = t_f$. These QOIs are a special case of Eq. (3.3), and we will discuss them below. The UQSA task is to compute estimates for the parametric sensitivity of Q , $\frac{dQ}{dp}$, as well as an estimate for the effect of different sources of error on the prediction of Q , denoted ΔQ .

3.1.1 Adjoint-based QOI sensitivity estimates

In this subsection, we derive the adjoint equations corresponding to system (3.1) and the adjoint-based expressions for computing $\frac{dQ}{dp}$. We begin by forming a Lagrangian, or *adjoined* QOI

$$\mathcal{L} = \int_{t_0}^{t_f} \left\{ \langle R(x, p, t) \rangle - \langle \lambda, F \rangle \right\} dt.$$

Note that if x satisfies $F(x, p, t) = 0$, then $Q = \mathcal{L}$ and $\frac{dQ}{dp} = \frac{d\mathcal{L}}{dp}$. The variable λ plays the role of a Lagrange multiplier and thus far is arbitrary; eventually we will place constraints on λ that will define the adjoint system of equations.

Before moving forward, we must define notation related to the inherent difference between the differential and algebraic parts of the system. First, λ naturally has differential and algebraic components, $\lambda = \langle \lambda^d, \lambda^a \rangle$, that correspond to the differential and algebraic components in x . Thus, the inner product $\langle \lambda, F \rangle$ may be more easily understood as

$$\langle \lambda, F \rangle = \langle \lambda^d, F^d \rangle + \langle \lambda^a, F^a \rangle. \quad (3.4)$$

Second, we give the chain rule in terms of the differential and algebraic components

of x :

$$\frac{dR}{dp} = \frac{\partial F}{\partial p} + \frac{\partial F}{\partial x^d} x_p^d + \frac{\partial F}{\partial x^a} x_p^a. \quad (3.5)$$

Application of this rule will determine the form of the differential and algebraic adjoint equations below.

To form a sensitivity equation, we take the functional derivative of the Lagrangian with respect to p . Using the chain rule, we find

$$\frac{d\mathcal{L}}{dp} = \int_{t_0}^{t_f} \left\{ \langle R_p \rangle + \langle R_x \rangle x_p - \frac{\partial}{\partial \dot{x}} \langle \lambda, F \rangle \dot{x}_p - \frac{\partial}{\partial x} \langle \lambda, F \rangle x_p - \frac{\partial}{\partial p} \langle \lambda, F \rangle \right\} dt.$$

We insert the following result from integration by parts,

$$\int_{t_0}^{t_f} \frac{\partial}{\partial \dot{x}} \langle \lambda, F \rangle \dot{x}_p dt = \left[\frac{\partial}{\partial \dot{x}} \langle \lambda, F \rangle x_p \right]_{t_0}^{t_f} - \int_{t_0}^{t_f} \frac{d}{dt} \left[\frac{\partial}{\partial \dot{x}} \langle \lambda, F \rangle \right] x_p dt,$$

and re-write the sensitivity equation as

$$\begin{aligned} \frac{d\mathcal{L}}{dp} &= - \left[\frac{\partial}{\partial \dot{x}} \langle \lambda, F \rangle x_p \right]_{t=t_0}^{t_f} + \int_{t_0}^{t_f} \left\{ \langle R_p \rangle + \langle R_x \rangle x_p + \frac{d}{dt} \left[\frac{\partial}{\partial \dot{x}} \langle \lambda, F \rangle \right] x_p \right. \\ &\quad \left. - \frac{\partial}{\partial x} \langle \lambda, F \rangle x_p - \frac{\partial}{\partial p} \langle \lambda, F \rangle \right\} dt \\ &= \left[\frac{\partial}{\partial \dot{x}} \langle \lambda, F \rangle x_p \right]_{t=t_0} - \left[\frac{\partial}{\partial \dot{x}} \langle \lambda, F \rangle x_p \right]_{t=t_f} + \int_{t_0}^{t_f} \left\{ \langle R_p \rangle - \frac{\partial}{\partial p} \langle \lambda, F \rangle \right\} dt \\ &\quad + \int_{t_0}^{t_f} \left\{ \left[\langle R_x \rangle + \frac{d}{dt} \left[\frac{\partial}{\partial \dot{x}} \langle \lambda, F \rangle \right] - \frac{\partial}{\partial x} \langle \lambda, F \rangle \right] x_p \right\} dt \end{aligned} \quad (3.6)$$

By inspection, the only term in Eq. (3.6) that we cannot evaluate or derive directly is x_p . This term is the full sensitivity of the solution vector with respect to all of the parameters for all time; if we knew this term, computing the sensitivity of derived QOIs would be a straightforward application of the chain rule. The operator

Jacobians, such as R_x , R_p , and F_p are straightforward (although often tedious) to compute given x and/or p .

The adjoint equations turn out to be the conditions that we impose on λ such that x_p is eliminated from the right hand side of Eq. (3.6). To see this condition, we re-write the integrand in the last term of Eq. (3.6) and manipulate using Eqs. (3.2), (3.4), and (3.5):

$$\begin{aligned} & \left[\langle R_x \rangle + \frac{d}{dt} \left[\frac{\partial}{\partial \dot{x}} \langle \lambda, F \rangle \right] - \frac{\partial}{\partial x} \langle \lambda, F \rangle \right] x_p \\ &= \left[\langle R_{x^d} \rangle + \frac{d}{dt} \langle \lambda^d \rangle - \frac{\partial}{\partial x^d} \left[\langle \lambda^d, F^d \rangle + \langle \lambda^a, F^a \rangle \right] \right] x_p^d \\ &+ \left[\langle R_{x^a} \rangle - \frac{\partial}{\partial x^a} \left[\langle \lambda^d, F^d \rangle + \langle \lambda^a, F^a \rangle \right] \right] x_p^a. \end{aligned}$$

From this we extract a differential and algebraic constraint, namely

$$\left\langle \frac{d\lambda^d}{dt} \right\rangle = \frac{\partial}{\partial x^d} \left[\langle \lambda^d, F^d \rangle + \langle \lambda^a, F^a \rangle \right] - \langle R_{x^d} \rangle \quad (3.7)$$

$$0 = \frac{\partial}{\partial x^a} \left[\langle \lambda^d, F^d \rangle + \langle \lambda^a, F^a \rangle \right] - \langle R_{x^a} \rangle, \quad (3.8)$$

that λ must satisfy in order to eliminate x_p^d and x_p^a , respectively, from Eq. (3.6). If these constraints are satisfied, the sensitivity equation reduces to

$$\begin{aligned} \frac{d\mathcal{L}}{dp} &= \left[\frac{\partial}{\partial \dot{x}} \langle \lambda, F \rangle x_p \right]_{t=t_0} - \left[\frac{\partial}{\partial \dot{x}} \langle \lambda, F \rangle x_p \right]_{t=t_f} + \int_{t_0}^{t_f} \left\{ \langle R_p \rangle - \frac{\partial}{\partial p} \langle \lambda, F \rangle \right\} dt \\ &= \left[\langle \lambda^d \rangle x_p^d \right]_{t=t_0} - \left[\langle \lambda^d \rangle x_p^d \right]_{t=t_f} + \int_{t_0}^{t_f} \left\{ \langle R_p \rangle - \frac{\partial}{\partial p} \langle \lambda, F \rangle \right\} dt. \end{aligned}$$

Note that x_p now appears at only the beginning and terminal time.

The term $x_p^d \Big|_{t=t_0}$ is simply the sensitivity of the initial conditions of the differential variables with respect to the parameters. This could have a wide range of meanings,

depending on the application. For example, if the initial condition is set by another code, x_p^d could be computed using a separate adjoint calculation within that code. For our purposes, we will consider the initial conditions as parameters themselves, meaning $\frac{dx^d}{dx_0} = 1$, providing an avenue for computing sensitivities with respect to initial conditions. In any case, we leave this term in the sensitivity equation and assume it is known or can be computed.

On the other hand, the value of x_p^d at $t = t_f$ is not known. We eliminate this term by setting an appropriate condition, known as the terminal condition, for the adjoint variable. The result is a system that must be solved backwards in time. The appropriate condition here is simply

$$\lambda^d(t_f) = 0. \quad (3.9)$$

The final adjoint initialization step is to compute $\lambda^a(t_f)$ using Eq. (3.8). The adjoint equations are solved backwards in time, and the forward and adjoint solutions are cross-correlated using the sensitivity equation

$$\begin{aligned} \frac{d\mathcal{L}}{dp} &= \left[\langle \lambda^d \rangle x_p^d \right]_{t=t_0} + \int_{t_0}^{t_f} \left\{ \langle R_p \rangle - \frac{\partial}{\partial p} \langle \lambda, F \rangle \right\} dt \\ &= \frac{dQ}{dp}. \end{aligned} \quad (3.10)$$

To review, the sensitivity equation, Eq. (3.10), is an expression for the first order total derivative of a time-integrated QOI with respect to a list of parameters, p . The expression requires a solution to the general forward problem, Eq. (3.1), and its corresponding adjoint problem, Eqs. (3.7)–(3.9). The advantage of the adjoint approach is that exactly one forward and one adjoint calculation are required to compute the full gradient, $\frac{dQ}{dp}$, no matter the dimension of p .

Many reactor analysis QOIs, such as nuclide inventories and system eigenvalues, are mainly of interest at $t = t_f$. Here we discuss a modification of the above formalism to account for this special case. We write the terminal QOI, Q_f , as

$$Q_f = \langle R(x, p, t_f) \rangle$$

The case of a terminal QOI is related to that of the integrated QOI by the following expression:

$$\frac{dQ_f}{dp} = \frac{d}{dt_f} \frac{dQ}{dp}. \quad (3.11)$$

We begin with Eq. (3.6), but account for our terminal condition, $\lambda^d(t_f) = 0$:

$$\begin{aligned} \frac{dQ}{dp} &= \left[\langle \lambda^d \rangle x_p^d \right]_{t=t_0} + \int_{t_0}^{t_f} \left\{ \langle R_p \rangle - \frac{\partial}{\partial p} \langle \lambda, F \rangle \right\} dt \\ &\quad + \int_{t_0}^{t_f} \left\{ \left[\langle R_x \rangle + \frac{d}{dt} \left[\frac{\partial}{\partial \dot{x}} \langle \lambda, F \rangle \right] - \frac{\partial}{\partial x} \langle \lambda, F \rangle \right] x_p \right\} dt. \end{aligned}$$

Applying rule (3.11) to this expression, we find

$$\begin{aligned}
\frac{dQ_f}{dp} &= \frac{d}{dt_f} \left[\left\langle \lambda^d \right\rangle x_p^d \right]_{t=t_0} + \left[\left\langle R_p \right\rangle - \frac{\partial}{\partial p} \left\langle \lambda, F \right\rangle \right]_{t=t_f} \\
&\quad + \left[\left(\left\langle R_x \right\rangle + \frac{d}{dt} \left[\frac{\partial}{\partial \dot{x}} \left\langle \lambda, F \right\rangle \right] - \frac{\partial}{\partial x} \left\langle \lambda, F \right\rangle \right) x_p \right]_{t=t_f} \\
&\quad - \int_{t_0}^{t_f} \frac{\partial}{\partial t_f} \left[\frac{\partial}{\partial p} \left\langle \lambda, F \right\rangle \right] dt \\
&\quad + \int_{t_0}^{t_f} \left\{ \left(\frac{\partial}{\partial t_f} \left[\frac{d}{dt} \left[\frac{\partial}{\partial \dot{x}} \left\langle \lambda, F \right\rangle \right] \right] - \frac{\partial}{\partial t_f} \left[\frac{\partial}{\partial x} \left\langle \lambda, F \right\rangle \right] \right) x_p \right\} dt \\
&= \left[\left\langle \lambda_f^d \right\rangle x_p^d \right]_{t=t_0} + \left[\left\langle R_p \right\rangle - \frac{\partial}{\partial p} \left\langle \lambda^a, F^a \right\rangle \right]_{t=t_f} \\
&\quad + \left[\left(\left\langle R_{x^d} \right\rangle + \frac{d}{dt} \left\langle \lambda^d \right\rangle - \frac{\partial}{\partial x^d} \left\langle \lambda^a, F^a \right\rangle \right) x_p^d \right]_{t=t_f} \\
&\quad + \left[\left(\left\langle R_{x^a} \right\rangle - \frac{\partial}{\partial x^a} \left\langle \lambda^a, F^a \right\rangle \right) x_p^a \right]_{t=t_f} \\
&\quad - \int_{t_0}^{t_f} \frac{\partial}{\partial p} \left\langle \lambda_f, F \right\rangle dt \\
&\quad + \int_{t_0}^{t_f} \left\{ \left(\frac{d}{dt} \left[\frac{\partial}{\partial \dot{x}} \left\langle \lambda_f, F \right\rangle \right] - \frac{\partial}{\partial x} \left\langle \lambda_f, F \right\rangle \right) x_p \right\} dt. \tag{3.12}
\end{aligned}$$

Here $\lambda_f \equiv \frac{\partial \lambda}{\partial t_f}$, as λ depends on both t and t_f . We take Eq. (3.12) term by term.

First, we extract adjoint equations from the last term in exactly the same manner as before. First we expand the integrand of the last term into differential and algebraic components

$$\begin{aligned}
&\left(\frac{d}{dt} \left[\frac{\partial}{\partial \dot{x}} \left\langle \lambda_f, F \right\rangle \right] - \frac{\partial}{\partial x} \left\langle \lambda_f, F \right\rangle \right) x_p \\
&= \left[\frac{d}{dt} \left\langle \lambda_f^d \right\rangle - \frac{\partial}{\partial x^d} \left[\left\langle \lambda_f^d, F^d \right\rangle + \left\langle \lambda_f^a, F^a \right\rangle \right] \right] x_p^d \\
&\quad + \left[- \frac{\partial}{\partial x^a} \left[\left\langle \lambda_f^d, F^d \right\rangle + \left\langle \lambda_f^a, F^a \right\rangle \right] \right] x_p^a, \tag{3.13}
\end{aligned}$$

and extract the following adjoint equations

$$\frac{d}{dt} \langle \lambda_f^d \rangle = \frac{\partial}{\partial x^d} \left[\langle \lambda_f^d, F^d \rangle + \langle \lambda_f^a, F^a \rangle \right] \quad (3.14)$$

$$\frac{\partial}{\partial x^a} \left[\langle \lambda_f^d, F^d \rangle + \langle \lambda_f^a, F^a \rangle \right] = 0. \quad (3.15)$$

Note that these are exactly the homogeneous parts of Eqs. (3.7) and (3.8), but that the adjoint variable carries a different mathematical interpretation. Next, the variable $\lambda^a(t_f)$ must satisfy

$$\frac{\partial}{\partial x^a} \langle \lambda^a, F^a \rangle = \langle R_{x^a} \rangle \quad (3.16)$$

to eliminate $x_p^a(t_f)$. The resulting value of $\lambda^a(t_f)$ appears in the second term of Eq. (3.12). The elimination of $x_d^p(t_f)$ leads to the appropriate terminal condition for $\lambda_f^d(t_f)$. Note that $\frac{d}{dt} \langle \lambda^d \rangle$ must satisfy

$$\frac{d}{dt} \langle \lambda^d \rangle = \frac{\partial}{\partial x^d} \langle \lambda^a, F^a \rangle - \langle R_{x^d} \rangle$$

in order to eliminate $x_p^d(t_f)$ from (3.12). Next, write our terminal condition for $\langle \lambda^d(t_f) \rangle$:

$$\langle \lambda^d(t_f) \rangle = 0$$

and take the total derivative with respect to time:

$$\left[\langle \lambda_f^d \rangle + \frac{d}{dt} \langle \lambda^d \rangle \right]_{t=t_f} = 0.$$

Then the terminal condition is simply

$$\langle \lambda_f^d(t_f) \rangle = -\frac{d}{dt} \langle \lambda^d(t_f) \rangle = \langle R_{x^d} \rangle - \frac{\partial}{\partial x^d} \langle \lambda^a, F^a \rangle. \quad (3.17)$$

Then $\lambda_f^a(t_f)$ is initialized by solving Eq. (3.15) at $t = t_f$. If λ_f satisfies Eqs. (3.14)–(3.15) with this initialization, the sensitivity of Q_f is given by

$$\frac{dQ_f}{dp} = \left[\langle \lambda_f^d \rangle x_p^d \right]_{t=t_0} + \left[\langle R_p \rangle - \frac{\partial}{\partial p} \langle \lambda^a, F^a \rangle \right]_{t=t_f} - \int_{t_0}^{t_f} \frac{\partial}{\partial p} \langle \lambda_f, F \rangle dt. \quad (3.18)$$

To review the case of a terminal QOI, we solve for an adjoint variable (λ_f) that is related to a time-derivative of the adjoint variable (λ). The resulting adjoint equations (Eqs. (3.14)–(3.15)) are the homogeneous version of those for an integrated QOI (Eqs. (3.7)–(3.8)), allowing for the use of the same software machinery. The terminal condition for λ_f is not trivially zero. Indeed, we will find that it requires an extra inversion of the algebraic equations.

Lastly, the terminal QOI sensitivity equation (Eq. (3.18)) contains an extra term involving λ^a evaluated at $t = t_f$. Note that if $R_{x^a} = 0$, or the QOI is not dependent on the algebraic variables, then $\lambda^a(t_f) = 0$, the $\frac{\partial}{\partial p} \langle \lambda^a, F^a \rangle$ term disappears from the sensitivity equation, and $\lambda_f^d(t_f) = R_{x^d}$.

3.1.2 Adjoint-based QOI error estimates

Another important result from our recent article[48] is an adjoint-based expression for estimating the global error in a QOI due to repeated or systematic local truncation errors. We showed that the same adjoint variable that is used to evaluate $\frac{dQ}{dp}$ can be used to propagate the local truncation errors to a global error estimate. Of course, to compute this estimate, the modeler must have a scheme for estimating these local truncation errors. Embedded Runge-Kutta schemes, described in the article and in

Sec. 2.3 of this dissertation, are one idea for estimating temporal discretization error. Schemes for estimating other truncation errors are available but not discussed in this work.

The error estimate extends the work of Cao and Petzold[47], which is related to ODEs, to the more general case of DAEs. Also, whenever possible, we maintain the general inner product notation to facilitate a range of choices for inner products and adjoint operators. In this subsection, use the following notation for the exact (continuous) system governing the unknowns:

$$F(\dot{x}, x, p, t) = 0, \quad x(t_0, p) = x_0(p). \quad (3.19)$$

We write the discrete system, or system that we actually are able to solve, as

$$F(\dot{\tilde{x}}, \tilde{x}, p, t) = r_1(t), \quad \tilde{x}(t_0, p) = x_0(p) + r_2, \quad (3.20)$$

where $r_1(t) = \begin{bmatrix} r_1^d(t) \\ r_1^a(t) \end{bmatrix}$ is a systematic, local perturbation resulting from a solution to the discrete problem instead of the exact problem, and r_2 is an error in the initial condition. These local errors certainly manifest as errors in the time series of the solution and therefore as predictive errors in any derived quantity of interest.

Here we consider the case of a terminal QOI. We quickly point out that this is not restrictive, as any time-integrated QOI can be transformed into a terminal QOI. To see this, consider the time-integrated QOI

$$Z = \int_{t_0}^{t_f} \langle R(x, p, t) \rangle dt.$$

Now consider a dummy variable z that satisfies $\frac{dz}{dt} = R(x, p, t)$. If we append z to the

forward system of equations with initial condition $z(t_0) = 0$ and integrate it along with the forward system, we find $Z = z(t_f)$. We could then apply the following formalism, treating $z(t)$ as an additional differential unknown.

The exact and approximate terminal QOIs are written as

$$Q = \langle R(x(t_f), p, t_f) \rangle$$

$$\tilde{Q} = \langle R(\tilde{x}(t_f), p, t_f) \rangle,$$

respectively. Thus, to $\mathcal{O}(\|(x - \tilde{x})^2\|) \equiv \mathcal{O}(\|e^2\|)$ -accuracy,

$$\Delta Q = Q - \tilde{Q} = \langle R_x, e(t_f) \rangle = \langle R_{x^d}, e^d(t_f) \rangle + \langle R_{x^a}, e^a(t_f) \rangle \quad (3.21)$$

Similarly, we linearize $F(\dot{x}, x, p, t)$ about $F(\dot{\tilde{x}}, \tilde{x}, p, t)$,

$$F(\dot{x}, x, p, t) = F(\dot{\tilde{x}}, \tilde{x}, p, t) + F_{\dot{x}}\dot{e} + F_x e + \mathcal{O}(\|e^2\|),$$

to write an $\mathcal{O}(\|e^2\|)$ -accurate equation satisfied by the error:

$$F_{\dot{x}}\dot{e} + F_x e = -r_1(t), \quad e(t_0, p) = r_2, \quad (3.22)$$

or

$$\begin{bmatrix} \frac{d\dot{e}}{dt} \\ 0 \end{bmatrix} = - \begin{bmatrix} F_{x^d}^d & F_{x^a}^d \\ F_{x^d}^a & F_{x^a}^a \end{bmatrix} \begin{bmatrix} e^d \\ e^a \end{bmatrix} - \begin{bmatrix} r_1^d \\ r_1^a \end{bmatrix}$$

$$e^d(t_0, p) = r_2^d \quad (3.23)$$

$$e^a(t_0, p) = r_2^a$$

This system of equations has an inherent linearization, which is why previously published analyses use the vector dot-product inner product to develop the error estimate. We proceed with a more general inner product. For either case, the analysis requires that $F_{x^a}^a$ be invertible, or that e^a can be solved for uniquely using the algebraic part of Eq. (3.23):

$$e^a = [F_{x^a}^a]^{-1} [-F_{x^d}^a e^d - r_1^a]. \quad (3.24)$$

All semi-explicit index-1 DAEs have this property. For example, in our depletion equations, the flux variable (and normalization, if applicable) are uniquely determined by the density unknowns. Therefore, we proceed under the assumption that we have this constraint available, and use it to substitute into Eq. (3.21) to form an expression for ΔQ in terms of $e^d(t_f)$ only:

$$\begin{aligned} \Delta Q &= \langle l_1, e^d(t_f) \rangle - \langle l_2 \rangle, \\ l_1 &= \left[R_{x^d} - R_{x^a} [F_{x^a}^a]^{-1} F_{x^d}^a \right]_{t=t_f} \\ l_2 &= \left[R_{x^a} [F_{x^a}^a]^{-1} r_1^a \right]_{t=t_f}. \end{aligned} \quad (3.25)$$

If we knew $e(t_f)$, we would be able to compute this estimate. Solving for $e(t_f)$ would require integrating Eq. (3.22) along the trajectory of \tilde{x} , which is at least equally expensive as integrating the governing equations and worse, would produce an error on the same order as the error that we are trying to predict! We find, however, with adjoint-differentiation techniques (see Appendix D) and the work by Cao & Petzold[47], we can obtain an estimate for $e(t_f)$ with the adjoint information we have already committed to computing.

To that end, we perform an index reduction by substituting the linear expression

for e^a into the differential components of Eq. (3.23). The result is an ODE for e^d :

$$\begin{aligned}\dot{e}^d &= \left[F_{x^a}^d [F_{x^a}^a]^{-1} F_{x^d}^a - F_{x^d}^d \right] e^d + F_{x^a}^d [F_{x^a}^a]^{-1} r_1^a - r_1^d, \\ e^d(t_0, p) &= r_2^d.\end{aligned}\quad (3.26)$$

We use a fundamental solution matrix, Φ , which satisfies

$$\dot{\Phi} = \left[F_{x^a}^d [F_{x^a}^a]^{-1} F_{x^d}^a - F_{x^d}^d \right] \Phi, \quad \Phi(t_0) = I,$$

to represent the solution, $e(t_f)$, according to the theory in Appendix D:

$$e^d(t_f) = \int_{t_0}^{t_f} \Phi(t_f) \Phi^{-1}(s) \left[F_{x^a}^d [F_{x^a}^a]^{-1} r_1^a(s) - r_1^d(s) \right] ds + \left[\Phi r_2^d \right]_{t=t_f}.$$

This allows us to write ΔQ in terms of Φ by substitution into Eq. (3.25):

$$\begin{aligned}\Delta Q &= \left\langle l_1, \int_{t_0}^{t_f} \Phi(t_f) \Phi^{-1}(s) \left[F_{x^a}^d [F_{x^a}^a]^{-1} r_1^a(s) - r_1^d(s) \right] ds + \left[\Phi r_2^d \right]_{t=t_f} \right\rangle - \left\langle l_2 \right\rangle \\ &= \int_{t_0}^{t_f} \left\langle l_1, \Phi(t_f) \Phi^{-1}(s) \left[F_{x^a}^d [F_{x^a}^a]^{-1} r_1^a(s) - r_1^d(s) \right] \right\rangle ds \\ &\quad + \left\langle l_1, \left[\Phi r_2^d \right]_{t=t_f} \right\rangle - \left\langle l_2 \right\rangle\end{aligned}\quad (3.27)$$

Now the trick is to relate l_1 and Φ to the adjoint variable that we have already committed to computing for the sensitivity analysis. We find that this is best illustrated using the specific case of the vector dot-product inner product because it provides explicit relations that are simple to follow. After the analysis, we show that the result would hold for the general inner product as well. In the case of the dot-product inner

product, the homogeneous part of system (3.22) has an associated adjoint equation

$$F_{\dot{x}} \dot{\lambda}_e = F_x^T \lambda_e, \quad (3.28)$$

for adjoint variable λ_e . After expanding this, we find

$$\begin{bmatrix} \frac{d\lambda_e^d}{dt} \\ 0 \end{bmatrix} = \begin{bmatrix} (F_{x^d}^d)^T & (F_{x^d}^a)^T \\ (F_{x^a}^d)^T & (F_{x^a}^a)^T \end{bmatrix} \begin{bmatrix} \lambda_e^d \\ \lambda_e^a \end{bmatrix}, \quad (3.29)$$

which gives a linear constraint for λ_e^a :

$$\lambda_e^a = - \left[(F_{x^a}^a)^T \right]^{-1} (F_{x^a}^d)^T \lambda_e^d. \quad (3.30)$$

If we perform another index reduction, by substituting this linear constraint into the differential equation for λ_e^d , we obtain the ODE

$$\frac{d\lambda_e^d}{dt} = - \left[F_{x^a}^d [F_{x^a}^a]^{-1} F_{x^d}^a - F_{x^d}^d \right]^T \lambda_e^d \quad (3.31)$$

Consider solving this ODE backwards in time with terminal condition

$$\lambda_e^d(t_f) = l_1^T \quad (3.32)$$

Comparing this ODE to that for e^d , Eq. (3.26), and using the result from Appendix D, we find that the solution $\lambda_e^d(t)$ can be related to Φ as follows:

$$(\lambda_e^d)^T(s) = l_1^T \Phi(t_f) \Phi^{-1}(s),$$

and $(\lambda_e^d)^T(0) = l_1^T \Phi(t_f)$. If we apply the dot-product inner product to Eq. (3.27)

and substitute this result, we find

$$\begin{aligned}
\Delta Q &= \int_{t_0}^{t_f} l_1^T \Phi(t_f) \Phi^{-1}(s) \left[F_{x^a}^d [F_{x^a}^a]^{-1} r_1^a(s) - r_1^d(s) \right] ds + l_1^T \Phi(t_f) r_2^d - l_2^T \\
&= \int_{t_0}^{t_f} [\lambda_e^d(s)]^T \left[F_{x^a}^d [F_{x^a}^a]^{-1} r_1^a(s) - r_1^d(s) \right] ds + [\lambda_e^d(0)]^T r_2^d - l_2^T \\
&= \int_{t_0}^{t_f} \left\{ -[\lambda_e^d(s)]^T r_1^d + [\lambda_e^d(s)]^T F_{x^a}^d [F_{x^a}^a]^{-1} r_1^a(s) \right\} ds + [\lambda_e^d(0)]^T r_2^d - l_2^T \\
&= \int_{t_0}^{t_f} -\left\{ [\lambda_e^d(s)]^T r_1^d(s) + [\lambda_e^a(s)]^T r_1^a(s) \right\} ds + [\lambda_e^d(0)]^T r_2^d - l_2^T. \quad (3.33)
\end{aligned}$$

Equation (3.33) gives the QOI error estimate in terms of the adjoint of the homogeneous error equation. We know or can compute all of the terms in this equation, except for possibly λ_e . Here we show that λ_e is the same variable as λ_f , the adjoint variable that we used to compute $\frac{dQ_f}{dp}$ in Sec. 3.1.1. First, we revisit the equations for λ_f using the dot-product inner product. The adjoint equations, Eqs. (3.14) and (3.15), are

$$\begin{aligned}
\frac{d}{dt} (\lambda_f^d)^T &= \frac{\partial}{\partial x^d} \left[(\lambda_f^d)^T F^d + (\lambda_f^a)^T F^a \right] \\
0 &= \frac{\partial}{\partial x^a} \left[(\lambda_f^d)^T F^d + (\lambda_f^a)^T F^a \right].
\end{aligned}$$

Note that the transpose of these equations is identical to Eq. (3.29); in other words, λ_e and λ_f satisfy the same differential-algebraic equations. To be identical, however, they must also share the same terminal condition. To get an explicit expression for $\lambda_f(t_f)$, we must solve for λ^a in Eq. (3.16). Using the dot-product inner product, we find

$$\frac{\partial}{\partial x^a} \left[(\lambda^a)^T F^a \right] = R_{x^a}$$

and therefore,

$$(\lambda^a)^T = R_{x^a} [F_{x^a}^a]^{-1}.$$

Substituting this expression into Eq. (3.17) we find the terminal condition for λ_f^d in the case of the dot-product inner product:

$$\begin{aligned} (\lambda_f^d)^T &= R_{x^d} - \frac{\partial}{\partial x^d} \left[R_{x^a} [F_{x^a}^a]^{-1} F^a \right] \\ &= R_{x^d} - R_{x^a} [F_{x^a}^a]^{-1} F_{x^d}^a \end{aligned}$$

Our terminal condition for λ_e^d was $\lambda_e^d(t_f) = l_1^T$ or

$$(\lambda_e^d)^T = l_1 = R_{x^d} - R_{x^a} [F_{x^a}^a]^{-1} F_{x^d}^a.$$

They are identical! The adjoint variables λ_e and λ_f are the same variables. This means that the expressions for the sensitivity of the QOI and the error in the QOI can be evaluated at the cost of a single adjoint solve.

Beginning with Eq. (3.28), we performed the analysis for the specific case of the vector dot-product inner product. This allowed us to write the explicit form of expressions (3.30), (3.31), and (3.33), which allowed for explicit substitutions that made the analysis easy to follow. We argue that the analysis will hold regardless of the inner product so long as the adjoint of the error equation, λ_e is equal to the adjoint previously developed for terminal QOIs, λ_f . To see this, we revisit the error equation. It was formulated by subtracting $F(\dot{\tilde{x}}, \tilde{x}, p, t)$ from

$$F(\dot{x}, x, p, t) = F(\dot{\tilde{x}}, \tilde{x}, p, t) + F_{\dot{x}} \dot{e} + F_x e + \mathcal{O}(|e^2|).$$

If we multiply this by the adjoint variable, we find to $\mathcal{O}(\|e^e\|)$:

$$\left\langle \lambda_e, F(\dot{x}, x, p, t) \right\rangle - \left\langle \lambda_e, F(\dot{\tilde{x}}, \tilde{x}, p, t) \right\rangle = \frac{\partial}{\partial \dot{x}} \left\langle \lambda_e, F(\dot{\tilde{x}}, \tilde{x}, p, t) \right\rangle \dot{e} + \frac{\partial}{\partial x} \left\langle \lambda_e, F(\dot{\tilde{x}}, \tilde{x}, p, t) \right\rangle e.$$

If we integrate the right hand side $t \in [t_0, t_f]$, and apply integration by parts to the first term, we find

$$\begin{aligned} & \int_{t_0}^{t_f} \left\{ \frac{\partial}{\partial \dot{x}} \left\langle \lambda_e, F(\dot{\tilde{x}}, \tilde{x}, p, t) \right\rangle \dot{e} + \frac{\partial}{\partial x} \left\langle \lambda_e, F(\dot{\tilde{x}}, \tilde{x}, p, t) \right\rangle e \right\} dt \\ &= \left[\frac{\partial}{\partial \dot{x}} \left\langle \lambda_e, F(\dot{\tilde{x}}, \tilde{x}, p, t) \right\rangle e \right]_{t=t_0}^{t=t_f} + \int_{t_0}^{t_f} \left\{ -\frac{d}{dt} \frac{\partial}{\partial \dot{x}} \left\langle \lambda_e, F(\dot{\tilde{x}}, \tilde{x}, p, t) \right\rangle e \right. \\ &\quad \left. + \frac{\partial}{\partial x} \left\langle \lambda_e, F(\dot{\tilde{x}}, \tilde{x}, p, t) \right\rangle e \right\} dt. \end{aligned}$$

Similar to before, we enforce the following adjoint equations in order to eliminate e from the integrand:

$$\frac{d}{dt} \frac{\partial}{\partial \dot{x}} \left\langle \lambda_e, F(\dot{\tilde{x}}, \tilde{x}, p, t) \right\rangle = \frac{\partial}{\partial x} \left\langle \lambda_e, F(\dot{\tilde{x}}, \tilde{x}, p, t) \right\rangle.$$

These are identical to Eq. (3.13). The adjoint equations corresponding to the error equation are the same as those corresponding to the governing equations no matter the form of the inner product. Also, because we have assumed $F_{x^a}^a$ is invertible, we will have analogs of Eqs. (3.30), (3.31), and (3.33); we just cannot write them here because their form will depend on the form of the inner product.

To review, we performed an analysis that provides an adjoint-based QOI error estimate, namely

$$\Delta Q = - \int_{t_0}^{t_f} \left\{ \left\langle \lambda, r_1 \right\rangle dt + \left\langle \lambda^d(t_0), r_2^d \right\rangle - \left\langle R_{x^a}, [F_{x^a}^a]^{-1} r_1^a \right\rangle_{t=t_f} \right\}. \quad (3.34)$$

Here, the adjoint variable λ is the same variable that was used to evaluate the sensitivity equation, Eq. (3.18).

3.1.3 A discussion of the adjoint formalism

In this subsection, we discuss a few observations about the adjoint formalism that we developed in the two preceding subsections. As a review, we derived equations that are adjoint to a forward system of equations that governs some physical process. We then showed that the solutions to these equations can be used to generate UQ information about quantities of interest that are derived from the forward equations. One such piece of information is the gradient of the QOI with respect to a list of parameters (possibly uncertain or design variables) that are input to the system. The second piece of information is an estimate of the global error in the predicted QOI, which results from the accumulation of local truncation errors incurred as the forward equations are being solved.

The first observation is that the adjoint equations are tightly coupled to the forward equations. The adjoint equations (either Eqs. (3.7) and (3.8) or Eqs. (3.14) and (3.15)) contain terms of the form $\frac{dF}{dx}$, which are linearizations about the trajectory of the forward solution. Therefore, in most cases, the forward solution will be required to compute these terms. Similar terms in the sensitivity equations (either Eq. (3.10) or Eq. (3.18)) and error equations (Eq. (3.34)) require access to both the forward and adjoint solutions, simultaneously. The bottom line is that the forward solution must be available in order to compute most terms in the adjoint equations and in the sensitivity expressions.

The forward and adjoint equations cannot be solved simultaneously, however, because the terminal adjoint condition ($\lambda(t_f)$) depends on the solution of the forward equations at $t = t_f$. Therefore, the procedure must be to solve the governing equa-

tions forward in time, set the terminal adjoint conditions at $t = t_f$, solve the adjoint equations backwards in time (while maintaining access to the forward solution), and somehow evaluate the sensitivity and error equations. This procedure poses challenging computational questions, many of which are addressed in this dissertation.

A second observation is that the adjoint variable plays the role of a weighting factor in the time integrals of the sensitivity and error equations. Indeed, the global sensitivity estimate takes the form of an adjoint-weighted integral of the local sensitivities; likewise, the global error estimate takes the form of an adjoint-weighted integral of the local truncation error estimates (plus some terms to account for initial conditions and errors propagated through the QOI expression). For this reason, the literature often refers the adjoint variable as an importance factor[27].

As an example application of this interpretation, consider the weighting of the local truncation error estimates to form a global error estimate. Suppose that the local truncation is constant in time, $r_1(t) = r_c$, but the magnitude of the adjoint variable increases by a factor of K as it is integrated from $t = t_f$ to $t = t_0$. This means that the truncation error made early in the simulation contributes to the global error with K times the importance. This will tell the modeler that the most effective efforts in error reduction (such as refinement) are those applied at early time steps.

A final observation is that our framework, or level of abstraction, is designed to provide flexibility in a modeling environment where different codes, models, and physics are being used to simulate the phenomenon of interest. The layer of abstraction provides a buffer between the modeler and the ground-level variational derivation of the adjoint equations, saving time in a situation where the modeler wants to exchange, add to, or otherwise modify the governing equations. He/she should be able to specialize the framework for the system of interest using general

adjoint equations that we have provided.

In the following three sections, we illustrate this benefit. In the first section, we apply our framework to the case of the source-driven depletion equations. We derive the corresponding adjoint equations for both a time-integrated and terminal QOI by simply interpreting the general adjoint equations laid out in this section. Then, in the second section, we repeat the derivation for the k-eigenvalue formulation of the depletion equations. In both cases, the time and effort to obtain the adjoint equations is much less than the full, from-scratch variational derivations provided in the literature to date. As a point of comparison, we provide these derivations in Appendix A. Both approaches yield the same equations. Finally, in the third section, we illustrate how the k-eigenvalue formulation of the adjoint depletion equations can be extended to include a forward and adjoint heat transfer equation.

Finally, our only assumption in this analysis was the index-1, semi-explicit formulation of the governing equations, which is common in engineering systems. The framework would also apply for systems of strictly differential equations (ODEs, for example, which are index-0 DAEs). The literature[19] provides adjoint information for other formulations of DAEs, but we find the index-1 form sufficient for the systems investigated in this dissertation.

As a review, our observations of the adjoint formalism are

1. Terms in the adjoint equations are linearized about the trajectory of the forward unknowns. Therefore, access to the forward equations is required while solving the adjoint equations.
2. The adjoint variable plays the role of a weighting factor. The global QOI sensitivity and error estimates turn out to be adjoint-weighted time integrals of local sensitivities and errors, respectively. In some sense, the adjoint variable

helps to propagate local estimates to global estimates.

3. The formalism we provide here remains as general as possible to facilitate a multi-physics modeling environment. The layer of abstraction provides a higher-level starting point for the modeler when changes to the physics model are necessary.
4. The formalism will apply for our formulation of the adjoint equations, as well as for most other reactor analysis systems of interest.

3.2 The Application of our Framework to the Source-Driven Depletion Equations

In Sec. 3.1, we derived a framework that provides the form of the adjoint equations corresponding to a general differential algebraic system. The framework also provides expressions for evaluating QOI sensitivities and errors. The entire analysis is written in terms of abstract variables and operators in order to facilitate the dynamics of a multi-physics modeling environment. We believe the abstraction is a powerful tool. Here, we illustrate its adaptability by specializing the framework for the case of the source-driven depletion equations.

We begin by writing the source driven depletion equations:

$$\frac{dN}{dt} - B(\psi, p)N = 0, \quad (3.35)$$

$$H(N, p)\psi - G(N, p)\psi - S_0 = 0, \quad (3.36)$$

$$N(t = 0) = N_0(p), \quad t \in [t_0, t_f]. \quad (3.37)$$

The abstraction of these equations to the general form of Eq. (3.1) is straightforward:

$$\begin{aligned} x^d &\leftarrow N \\ x^a &\leftarrow \psi \\ \lambda^d &\leftarrow N^\dagger \\ \lambda^a &\leftarrow \psi^\dagger \\ F^d &\leftarrow \frac{dN}{dt} - BN \\ F^a &\leftarrow H\psi - G\psi - S_0 \end{aligned}$$

Note that all operators and the initial conditions are generally dependent on the

vector of parameters, p . As in Eq. (3.3), we write our time-integrated QOI as

$$Q = \int_{t_0}^{t_f} \left\langle R(N, \psi, p, t) \right\rangle_{E, \mathcal{D}, \Omega} dt, \quad (3.38)$$

except we define a specific inner product as the integration over volume, angle, and energy:

$$\left\langle \cdot \right\rangle_{E, \mathcal{D}, \Omega} = \int dE \int d\Omega \int dV.$$

Recall the adjoint transport operators that we developed in Sec. 2.4:

$$\left\langle \psi^\dagger, H\psi \right\rangle_{E, \mathcal{D}, \Omega} = \left\langle \psi, H^\dagger \psi^\dagger \right\rangle_{E, \mathcal{D}, \Omega}, \quad (3.39)$$

$$\left\langle \psi^\dagger, G\psi \right\rangle_{E, \mathcal{D}, \Omega} = \left\langle \psi, G^\dagger \psi^\dagger \right\rangle_{E, \mathcal{D}, \Omega}, \quad (3.40)$$

Also in Sec. 2.4, we derived a form of the Bateman equation that was written in terms of this same phase-space inner product, namely

$$\frac{dN}{dt} - \left\langle bN \right\rangle_{E, \mathcal{D}, \Omega} = 0$$

where

$$\left\langle bN \right\rangle_{E, \mathcal{D}, \Omega} = BN.$$

Recall that this formulation assumes that we use spatially cell-averaged nuclide densities as the density unknown. The adjoint Bateman operator is

$$\left\langle N^\dagger, bN \right\rangle_{E, \mathcal{D}, \Omega} = \left\langle N, b^\dagger N^\dagger \right\rangle_{E, \mathcal{D}, \Omega}. \quad (3.41)$$

For the time-integrated QOI, we found adjoint equations (3.7) and (3.8), which

we re-write here in terms of our inner product:

$$\begin{aligned} \left\langle \frac{dN^\dagger}{dt} \right\rangle_{E,\mathcal{D},\Omega} &= \frac{\partial}{\partial N} \left[\left\langle N^\dagger, \frac{dN}{dt} - BN \right\rangle_{E,\mathcal{D},\Omega} \right. \\ &\quad \left. + \left\langle \psi^\dagger, H\psi - G\psi - S_0 \right\rangle_{E,\mathcal{D},\Omega} \right] - \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} \\ 0 &= \frac{\partial}{\partial \psi} \left[\left\langle N^\dagger, \frac{dN}{dt} - BN \right\rangle_{E,\mathcal{D},\Omega} \right. \\ &\quad \left. + \left\langle \psi^\dagger, H\psi - G\psi - S_0 \right\rangle_{E,\mathcal{D},\Omega} \right] - \left\langle \frac{\partial R}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega}. \end{aligned}$$

We immediately note that the form of these equations is odd because the Bateman variables and operators are not dependent on angle and energy. To that end, define a temporary variable

$$\tilde{N}^\dagger = \left\langle N^\dagger \right\rangle_{E,\mathcal{D},\Omega} = \left\langle \right\rangle_{E,\Omega} N^\dagger V_c, \quad (3.42)$$

where V_c is a cell volume over which N^\dagger has been averaged. Using this notation and the following manipulations

$$\begin{aligned} \frac{\partial}{\partial N} \left\langle N^\dagger, \frac{dN}{dt} - BN \right\rangle_{E,\mathcal{D},\Omega} &= - \left\langle N^\dagger, \frac{\partial}{\partial N} BN \right\rangle_{E,\mathcal{D},\Omega} = -B^\dagger \tilde{N}^\dagger \\ \frac{\partial}{\partial N} \left\langle \psi^\dagger, H\psi - G\psi - S_0 \right\rangle_{E,\mathcal{D},\Omega} &= \left\langle \psi^\dagger, \frac{\partial H\psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial G\psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} \\ \frac{\partial}{\partial \psi} \left\langle N^\dagger, \frac{dN}{dt} - BN \right\rangle_{E,\mathcal{D},\Omega} &= \frac{\partial}{\partial \psi} \left[\tilde{N}^\dagger \left(\frac{dN}{dt} - BN \right) \right] \\ &= -\tilde{N}^\dagger \frac{\partial BN}{\partial \psi} = - \left\langle \tilde{N}^\dagger \frac{\partial BN}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} \\ \frac{\partial}{\partial \psi} \left\langle \psi^\dagger, H\psi - G\psi - S_0 \right\rangle_{E,\mathcal{D},\Omega} &= \frac{\partial}{\partial \psi} \left\langle \psi, H^\dagger \psi^\dagger - G^\dagger \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} = \left\langle H^\dagger \psi^\dagger - G^\dagger \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} \end{aligned}$$

we re-write the adjoint equations as

$$\begin{aligned}\frac{d\tilde{N}^\dagger}{dt} &= -B^\dagger \tilde{N}^\dagger + \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - G\psi] \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} \\ 0 &= \left\langle H^\dagger \psi^\dagger - G^\dagger \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \tilde{N}^\dagger \frac{\partial bN}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\partial R}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega}.\end{aligned}$$

These equations have a more intuitive connection to the forward equations. Moving forward, we drop the \tilde{N}^\dagger notation in favor of N^\dagger . Lastly, we note that all terms in the second equation could be combined into a single inner product. The final source driven adjoint equations for a time integrated QOI are

$$\frac{dN^\dagger}{dt} = -B^\dagger N^\dagger + \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - G\psi] \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} \quad (3.43)$$

$$H^\dagger \psi^\dagger - G^\dagger \psi^\dagger = N^\dagger \frac{\partial bN}{\partial \psi} + \frac{\partial R}{\partial \psi}. \quad (3.44)$$

By (3.9), our terminal condition for N^\dagger is

$$N^\dagger(t_f) = 0, \quad (3.45)$$

and, by Eq. (3.44), $\psi^\dagger(t_f)$ is determined by solving

$$H^\dagger \psi^\dagger - G^\dagger \psi^\dagger = \frac{\partial R}{\partial \psi}$$

with all terms evaluated at $t = t_f$. If this system is solved for N^\dagger and ψ^\dagger , then the

sensitivity equation, Eq. (3.10), for our case becomes

$$\begin{aligned} \frac{dQ}{dp} = & \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0} + \int_{t_0}^{t_f} \left\{ \left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - G\psi - S_0] \right\rangle_{E,\mathcal{D},\Omega} \right. \\ & \left. + \left\langle N^\dagger, \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \right\} dt. \end{aligned} \quad (3.46)$$

In summary, for the source-driven forward problem Eqs. (3.35) – (3.37) with time-integrated QOI of the form Eq. (3.38), the corresponding adjoint equations are Eqs. (3.43) and (3.44) with terminal condition given by Eq. (3.45). Given solutions to the forward and adjoint problem, the gradient of the QOI with respect to the list of parameters p is given by Eq. (3.46).

For the case of a terminal solution QOI of the form

$$Q_f = \left\langle R(N, \psi, p, t_f) \right\rangle_{E,\mathcal{D},\Omega}, \quad (3.47)$$

we found in Sec. 3.1.1 that the adjoint equations are simply the homogeneous part of Eqs. (3.43) and (3.44). That is,

$$\frac{dN^\dagger}{dt} = \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - G\psi] \right\rangle_{E,\mathcal{D},\Omega} - B^\dagger N^\dagger \quad (3.48)$$

$$H^\dagger \psi^\dagger - G^\dagger \psi^\dagger = N^\dagger \frac{\partial bN}{\partial \psi}. \quad (3.49)$$

We also found that the initialization of the adjoint variables is not as trivial in the case of a terminal QOI. We first define an intermediate flux variable, $\hat{\psi}^\dagger$, which corresponds to λ^a in Eq. (3.16). We find that Eq. (3.16) corresponds to

$$\frac{d}{d\psi} \left\langle \hat{\psi}^\dagger, H\psi - G\psi - S_0 \right\rangle_{E,\mathcal{D},\Omega} = \left\langle H^\dagger \hat{\psi}^\dagger - G^\dagger \hat{\psi}^\dagger \right\rangle_{E,\mathcal{D},\Omega} = \left\langle \frac{\partial R}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega}$$

or

$$H^\dagger \hat{\psi}^\dagger - G^\dagger \hat{\psi}^\dagger = \frac{\partial R}{\partial \psi} \quad (3.50)$$

which must be solved for $\hat{\psi}^\dagger$ with all terms evaluated at $t = t_f$. Then, by Eq. (3.17), the adjoint density terminal condition is

$$N^\dagger(t_f) = \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\partial}{\partial N} [H\psi - G\psi], \hat{\psi}^\dagger \right\rangle_{E,\mathcal{D},\Omega}. \quad (3.51)$$

Given the solution to Eqs. (3.48) and (3.49), with terminal condition Eq. (3.51), we find that Eq. (3.18), the sensitivity expression for Q_f , translates to

$$\begin{aligned} \frac{dQ_f}{dp} &= \left[\left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \hat{\psi}^\dagger, \frac{\partial}{\partial p} [H\psi - G\psi - S_0] \right\rangle_{E,\mathcal{D},\Omega} \right]_{t=t_f} + \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0} \\ &\quad - \int_{t_0}^{t_f} \left\{ \left\langle \hat{\psi}^\dagger, \frac{\partial}{\partial p} [H\psi - G\psi - S_0] \right\rangle_{E,\mathcal{D},\Omega} - \left\langle N^\dagger, \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \right\} dt \end{aligned} \quad (3.52)$$

In summary, if the depletion problem is source-driven, and the QOI is a functional of the terminal forward solution only, the corresponding adjoint equations are Eqs. (3.48) – (3.49) with terminal condition given by Eq. (3.51). The gradient of the QOI is then given by (3.52). If the QOI depends on the flux, the terminal condition for N^\dagger comes at a cost of one transport solve, given by Eq. (3.50). If the QOI does not depend on the flux (e.g. a terminal nuclide density), then $\hat{\psi} = 0$, removing the cost of this transport solve as well as the second term in Eq. (3.52).

We also have an adjoint-based estimate for the manifestation of local truncation errors in the predicted, terminal QOI. This expression uses the same adjoint variable as the sensitivity equation, Eq. (3.52), so we do not need to re-derive and compute a

new adjoint solution. We are required, however, to compute local truncation errors, for example due to discretization, during the forward solve. Denote these estimates as $r_N(t)$ for the Bateman equations and $r_\psi(t)$ for the transport equation. We might also have an estimate for the error in the initial nuclide densities; denote this error r_{N_0} . Then Eq. (3.34) gives the global error estimate as

$$\begin{aligned}\Delta Q_f = & - \int_{t_0}^{t_f} \left\{ N^\dagger \cdot r_N + \left\langle \psi^\dagger, r_\psi \right\rangle_{E, \mathcal{D}, \Omega} \right\} dt \\ & + N^\dagger(t_0) \cdot r_{N_0} - \left\langle r_\psi(t_f), \hat{\psi}^\dagger \right\rangle_{E, \mathcal{D}, \Omega}. \end{aligned}\quad (3.53)$$

3.3 The Application of our Framework to the k-Eigenvalue Depletion Equations

In this section, we apply the framework described in Sec. 3.1 to derive the adjoint equations corresponding to the k-eigenvalue formulation of the burnup equations. In the previous section, we applied the framework to the source-driven depletion formulation. Thus, this section works to illustrate the flexibility afforded by the general framework's abstraction. Instead of going back to the variational derivation to derive the adjoint equations for the k-eigenvalue case, we are able to simply re-translate the equations provided by the framework.

We begin the analysis by writing the k-eigenvalue analog to Eqs. (3.35) – (3.37) but with the addition of a power constraint:

$$\frac{dN}{dt} - B(\psi, A, p)N = 0, \quad (3.54)$$

$$H(N, p)\psi - \lambda G(N, p)\psi = 0, \quad (3.55)$$

$$AP(N, p)\psi - \mathcal{P}(t) = 0 \quad (3.56)$$

$$N(t = 0) = N_0(p), \quad t \in [t_o, t_f]. \quad (3.57)$$

Here, ψ satisfies the fundamental eigenmode of Eq. (3.55) with eigenvalue $\lambda = \frac{1}{k_{\text{eff}}}$, and the parameter $\mathcal{P}(t)$ is the prescribed average power density ($[\frac{\text{W}}{\text{cm}^3}]$) of the reactor, which is a constraint on the system. The operator P is used to compute the average

power density generated by fission,

$$\begin{aligned}
\text{Power Density} &= AP\psi \equiv A \int dV \int dE \int d\Omega \frac{\sum_j N_j(r) \sigma_{f,j}(E) E_{f,j} \psi(r, E, \Omega)}{V_R} \\
&= \frac{A}{V_R} \left\langle \sum_j N_j \sigma_{f,j} E_{f,j} \psi \right\rangle_{E, \mathcal{D}, \Omega} \\
&\equiv \frac{A}{V_R} \left\langle \Sigma_E \psi \right\rangle_{E, \mathcal{D}, \Omega} \\
&\equiv \frac{A}{V_R} \hat{\mathcal{P}}
\end{aligned}$$

where $E_{f,j}$ is the energy released in a fission event by nuclide j , V_R is the reactor volume, and the symbols Σ_E and $\hat{\mathcal{P}}$ will be used as a short-hand for an energy-production macroscopic cross section and un-normalized power, respectively. Finally, the scalar normalization factor A is the value by which the eigenvector must be scaled to satisfy the power constraint.

Before proceeding with the derivation of the adjoint equations, we must note a subtle difference in the form of the Bateman operator. Because the magnitude of the solution ψ to Eq. (3.55) is arbitrary, the operator B must be a function of the normalization constant, A . We know, however, that the dependency will be bilinear in A and ψ ; that is, these variables will only appear as a product together. To that end, define $\Psi = A\psi$, and note the following relationships that will serve as a convenience in this section:

$$\frac{\partial B}{\partial \psi} = \frac{\partial B}{\partial \Psi} \frac{\partial \Psi}{\partial \psi} = \frac{\partial B}{\partial \Psi} A \quad (3.58)$$

$$\frac{\partial B}{\partial A} = \frac{\partial B}{\partial \Psi} \frac{\partial \Psi}{\partial A} = \frac{\partial B}{\partial \Psi} \psi. \quad (3.59)$$

Equations (3.54) – (3.57) form a DAE system where the algebraic constraint is an eigenvalue equation accompanied by a normalization for the eigenvector. Thus,

we can apply our framework given in Sec. 3.1. The abstraction of the governing equations is as follows:

$$\begin{aligned}
x^d &\leftarrow N \\
x^a &\leftarrow \begin{bmatrix} \psi/\lambda \\ A \end{bmatrix} \\
\lambda^d &\leftarrow N^\dagger \\
\lambda^a &\leftarrow \begin{bmatrix} \psi^\dagger \\ A^\dagger \end{bmatrix} \\
F^d &\leftarrow \frac{dN}{dt} - BN \\
F^a &\leftarrow \begin{bmatrix} H\psi - \lambda G\psi \\ AP\psi - \mathcal{P}(t) \end{bmatrix}
\end{aligned}$$

Note that both ψ and λ are listed as algebraic variables, as they, as a set, form the solution to the eigenvalue equation.

Proceeding as before, we first define a time-integrated QOI as

$$Q = \int_{t_0}^{t_f} \left\langle R(N, \psi, A, p, t) \right\rangle_{E, \mathcal{D}, \Omega} dt. \quad (3.60)$$

Note that we write the QOI dependence on both ψ and A , although in most normal cases they will appear together. We did not, however, write a dependence on λ , because a time integral of the eigenvalue is generally not a useful QOI. Our initial

translation of Eqs. (3.7) and (3.8) are

$$\begin{aligned} \left\langle \frac{dN^\dagger}{dt} \right\rangle_{E,\mathcal{D},\Omega} &= \frac{\partial}{\partial N} \left\langle N^\dagger, \frac{dN}{dt} - BN \right\rangle_{E,\mathcal{D},\Omega} + \frac{\partial}{\partial N} \left\langle \psi^\dagger, H\psi - \lambda G\psi \right\rangle_{E,\mathcal{D},\Omega} \\ &\quad + \frac{\partial}{\partial N} \left\langle A^\dagger, \frac{A}{V_R} \hat{\mathcal{P}} - \mathcal{P} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} \end{aligned} \quad (3.61)$$

$$\begin{aligned} 0 &= \frac{\partial}{\partial \psi} \left\langle N^\dagger, \frac{dN}{dt} - BN \right\rangle_{E,\mathcal{D},\Omega} + \frac{\partial}{\partial \psi} \left\langle \psi^\dagger, H\psi - \lambda G\psi \right\rangle_{E,\mathcal{D},\Omega} \\ &\quad + \frac{\partial}{\partial \psi} \left\langle A^\dagger, \frac{A}{V_R} \hat{\mathcal{P}} - \mathcal{P} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\partial R}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} \end{aligned} \quad (3.62)$$

$$\begin{aligned} 0 &= \frac{\partial}{\partial \lambda} \left\langle N^\dagger, \frac{dN}{dt} - BN \right\rangle_{E,\mathcal{D},\Omega} + \frac{\partial}{\partial \lambda} \left\langle \psi^\dagger, H\psi - \lambda G\psi \right\rangle_{E,\mathcal{D},\Omega} \\ &\quad + \frac{\partial}{\partial \lambda} \left\langle A^\dagger, \frac{A}{V_R} \hat{\mathcal{P}} - \mathcal{P} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\partial R}{\partial \lambda} \right\rangle_{E,\mathcal{D},\Omega} \end{aligned} \quad (3.63)$$

$$\begin{aligned} 0 &= \frac{\partial}{\partial A} \left\langle N^\dagger, \frac{dN}{dt} - BN \right\rangle_{E,\mathcal{D},\Omega} + \frac{\partial}{\partial A} \left\langle \psi^\dagger, H\psi - \lambda G\psi \right\rangle_{E,\mathcal{D},\Omega} \\ &\quad + \frac{\partial}{\partial A} \left\langle A^\dagger, \frac{A}{V_R} \hat{\mathcal{P}} - \mathcal{P} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} \end{aligned} \quad (3.64)$$

We insert an equation for both ψ and λ because they are each members of the algebraic variable vector and are each sensitive to perturbations in p . We perform the same change of variable, given by Eq. (3.42). Also, because A^\dagger is a scalar, we define

$$\tilde{A}^\dagger = \left\langle A^\dagger \right\rangle_{E,\mathcal{D},\Omega} = A^\dagger \left\langle \right\rangle_{E,\mathcal{D},\Omega}$$

We again take these equations term by term. Starting with Eq. (3.61):

$$\begin{aligned} \frac{\partial}{\partial N} \left\langle N^\dagger, \frac{dN}{dt} - BN \right\rangle_{E,\mathcal{D},\Omega} &= - \left\langle N^\dagger, \frac{\partial}{\partial N} BN \right\rangle_{E,\mathcal{D},\Omega} = -B^\dagger \tilde{N}^\dagger \\ \frac{\partial}{\partial N} \left\langle \psi^\dagger, H\psi - \lambda G\psi - S_0 \right\rangle_{E,\mathcal{D},\Omega} &= \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \\ \frac{\partial}{\partial N} \left\langle A^\dagger, \frac{A}{V_R} \hat{\mathcal{P}} - \mathcal{P} \right\rangle_{E,\mathcal{D},\Omega} &= \frac{\partial}{\partial N} \left[\tilde{A}^\dagger \left[\frac{A}{V_R} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} - \mathcal{P} \right] \right] \\ &= \tilde{A}^\dagger \frac{A}{V_R} \frac{\partial \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega}}{\partial N} = \left\langle \tilde{A}^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} \end{aligned}$$

From this we find the adjoint Bateman equation

$$\frac{d\tilde{N}^\dagger}{dt} = -B^\dagger \tilde{N}^\dagger + \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \tilde{A}^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega}.$$

Now for the terms in Eq. (3.62):

$$\begin{aligned} \frac{\partial}{\partial \psi} \left\langle N^\dagger, \frac{dN}{dt} - BN \right\rangle_{E,\mathcal{D},\Omega} &= -\frac{\partial}{\partial \psi} \left[\tilde{N}^\dagger \left(\frac{dN}{dt} - BN \right) \right] \\ &= -\tilde{N}^\dagger \frac{\partial BN}{\partial \psi} = -\left\langle \tilde{N}^\dagger \frac{\partial bN}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} \\ \frac{\partial}{\partial \psi} \left\langle \psi^\dagger, H\psi - \lambda G\psi - S_0 \right\rangle_{E,\mathcal{D},\Omega} &= \frac{\partial}{\partial \psi} \left\langle \psi, H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} \\ &= \left\langle H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} \\ \frac{\partial}{\partial \psi} \left\langle A^\dagger, \frac{A}{V_R} \hat{\mathcal{P}} - \mathcal{P} \right\rangle_{E,\mathcal{D},\Omega} &= \frac{\partial}{\partial \psi} \left[\tilde{A}^\dagger \left[\frac{A}{V_R} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} - \mathcal{P} \right] \right] \\ &= \tilde{A}^\dagger \frac{A}{V_R} \left\langle \Sigma_E \right\rangle_{E,\mathcal{D},\Omega} = \left\langle \tilde{A}^\dagger \frac{A}{V} \Sigma_E \right\rangle_{E,\mathcal{D},\Omega} \end{aligned}$$

We note that all terms may be combined into a single inner product, and the resulting adjoint transport equation is

$$H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger = S^\dagger$$

with

$$S^\dagger = \frac{\partial R}{\partial \psi} + \tilde{N}^\dagger \frac{\partial bN}{\partial \psi} - \tilde{A}^\dagger \frac{A}{V_R} \Sigma_E.$$

Per the definition of λ , the operator on the left hand side of the adjoint transport equation, $H^\dagger - \lambda G^\dagger$, is singular. This brings the solvability of this equation into question, and we will address this below. For Eq. (3.63), we find that the only non-zero homogeneous term is $\frac{\partial}{\partial \lambda} \left\langle \psi^\dagger, \lambda G\psi \right\rangle_{E,\mathcal{D},\Omega}$. The partials with respect to all other

transport and Bateman operators are zero. The integrated QOI $Q = \int_{t_0}^{t_f} \langle \lambda \rangle_{E,\mathcal{D},\Omega} dt$ is generally not of any interest to reactor analysts, so we assume that $\frac{\partial R}{\partial \lambda} = 0$.

Therefore, the constraint we get out of this equation is

$$\langle \psi^\dagger, G\psi \rangle_{E,\mathcal{D},\Omega} = 0,$$

which simply says that the adjoint flux must be orthogonal to the forward fission source. More detail on the mathematical implications of this constraint are discussed in Sec. 4.4. Finally, the non-zero terms in Eq. (3.64) are

$$\begin{aligned} \frac{\partial}{\partial A} \left\langle N^\dagger, \frac{dN}{dt} - BN \right\rangle_{E,\mathcal{D},\Omega} &= -\frac{\partial}{\partial A} \left[\tilde{N}^\dagger \left(\frac{dN}{dt} - BN \right) \right] \\ &= -\tilde{N}^\dagger \frac{\partial BN}{\partial A} = -\left\langle \tilde{N}^\dagger \frac{\partial bN}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} \\ \frac{\partial}{\partial A} \left\langle A^\dagger, \frac{A}{V_R} \hat{\mathcal{P}} - \mathcal{P} \right\rangle_{E,\mathcal{D},\Omega} &= \tilde{A}^\dagger \frac{\hat{\mathcal{P}}}{V_R}. \end{aligned}$$

Leading to a simple expression for \tilde{A}^\dagger :

$$\tilde{A}^\dagger = \frac{V_R}{\hat{\mathcal{P}}} \left[\left\langle \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \tilde{N}^\dagger \frac{\partial bN}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} \right]$$

We again drop the \tilde{N}^\dagger and \tilde{A}^\dagger notation for N^\dagger and A^\dagger , and find that our k-eigenvalue

adjoint depletion equations are

$$\begin{aligned} \frac{dN^\dagger}{dt} &= -B^\dagger N^\dagger + \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \\ &\quad + \left\langle A^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} \end{aligned} \quad (3.65)$$

$$H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger = \frac{\partial R}{\partial \psi} + N^\dagger \frac{\partial bN}{\partial \psi} - A^\dagger \frac{A}{V_R} \Sigma_E \quad (3.66)$$

$$\left\langle \psi^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} = 0 \quad (3.67)$$

$$A^\dagger = \frac{V_R}{\hat{\mathcal{P}}} \left[\left\langle \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle N^\dagger \frac{\partial bN}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} \right] \quad (3.68)$$

The terminal condition for the adjoint densities is again

$$N^\dagger(t_f) = 0, \quad (3.69)$$

and the terminal values for ψ^\dagger and A^\dagger are computed with Eqs. (3.66) and (3.68), respectively, subject to (3.67). Once this system is solved, the k-eigenvalue translation of Eq. (3.10) gives the sensitivity of the QOI:

$$\begin{aligned} \frac{dQ}{dp} &= \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0} + \int_{t_0}^{t_f} \left\{ \left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right. \\ &\quad \left. + \left\langle N^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} + A^\dagger \frac{\partial \mathcal{P}(t)}{\partial p} \right\} dt. \end{aligned} \quad (3.70)$$

We now address the solvability of the adjoint transport equation. The Fredholm alternative theorem[50] requires that for $H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger = S^\dagger$ to have a unique solution, the following must be true:

$$\left\langle S^\dagger, \psi \right\rangle_{E,\mathcal{D},\Omega} = 0, \quad (3.71)$$

where ψ is the solution to $H\psi - \lambda G\psi = 0$, the forward equation. To see the criteria under which this condition holds, we expand Eq. (3.71) and use Eq. (3.58) as follows:

$$\begin{aligned}\left\langle S^\dagger, \psi \right\rangle_{E,\mathcal{D},\Omega} &= \left\langle \frac{\partial R}{\partial \psi}, \psi \right\rangle_{E,\mathcal{D},\Omega} + \left\langle N^\dagger \frac{\partial bN}{\partial \psi}, \psi \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A^\dagger \frac{A}{V_R} \Sigma_E, \psi \right\rangle_{E,\mathcal{D},\Omega} \\ &= \left\langle \frac{\partial R}{\partial \psi}, \psi \right\rangle_{E,\mathcal{D},\Omega} + \left\langle AN^\dagger \frac{\partial bN}{\partial \Psi}, \psi \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A^\dagger \frac{A}{V_R} \Sigma_E, \psi \right\rangle_{E,\mathcal{D},\Omega}. \quad (3.72)\end{aligned}$$

Next we multiply Eq. (3.68) by A , use Eq. (3.59), and manipulate to solve for $\left\langle AN^\dagger \frac{\partial bN}{\partial \Psi}, \psi \right\rangle_{E,\mathcal{D},\Omega}$:

$$\begin{aligned}AA^\dagger &= \frac{V_R}{\mathcal{P}} \left[\left\langle A \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle AN^\dagger \frac{\partial bN}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} \right] \\ &= \frac{V_R}{\left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega}} \left[\left\langle A \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle AN^\dagger \frac{\partial bN}{\partial \Psi} \psi \right\rangle_{E,\mathcal{D},\Omega} \right] \\ &\rightarrow \left\langle AN^\dagger \frac{\partial bN}{\partial \Psi} \psi \right\rangle_{E,\mathcal{D},\Omega} = \left\langle \frac{AA^\dagger \Sigma_E \psi}{V_R} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega}\end{aligned}$$

Substitution of this result into Eq. (3.72), we find that $\left\langle S^\dagger, \psi \right\rangle_{E,\mathcal{D},\Omega} = 0$ is satisfied if our QOI satisfies

$$\left\langle \frac{\partial R}{\partial \psi}, \psi \right\rangle_{E,\mathcal{D},\Omega} = \left\langle A \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega}. \quad (3.73)$$

This condition is met by any function R that is bilinear in A and ψ or is a ratio of bilinear functions of A and ψ . In other words, A and ψ should appear as a product. Because virtually any practical QOI depends on the normalized flux, this condition will nearly always be met. Nonetheless, it must be considered.

We next turn to the case of a terminal QOI. The QOI expression is

$$Q_f = \left\langle R(N, \psi, A, \lambda, p, t_f) \right\rangle_{E,\mathcal{D},\Omega}. \quad (3.74)$$

Here we include λ as an argument, as the QOI could be the end-of-life eigenvalue. As before, the adjoint equations are simply the homogeneous parts of Eqs. (3.65)–(3.68), namely

$$\frac{dN^\dagger}{dt} = -B^\dagger N^\dagger + \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} + \left\langle A^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} \quad (3.75)$$

$$H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger = N^\dagger \frac{\partial bN}{\partial \psi} - A^\dagger \frac{A}{V_R} \Sigma_E \quad (3.76)$$

$$\left\langle \psi^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} = 0 \quad (3.77)$$

$$A^\dagger = \frac{V_R}{\hat{\mathcal{P}}} \left\langle N^\dagger \frac{\partial bN}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} \quad (3.78)$$

We know from the previous analysis that the right hand side of Eq. (3.76) is orthogonal to ψ . Moving to the terminal condition, we first define two intermediate quantities, $\hat{\psi}^\dagger$ and \hat{A}^\dagger that correspond to λ^a in Eq. (3.16). We then expand Eq. (3.16) for each of our three algebraic variables, accounting for the fact that $\frac{dR}{d\lambda}$ may be non-zero:

$$\begin{aligned} \left\langle \frac{\partial R}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} &= \frac{\partial}{\partial \psi} \left\langle \hat{\psi}^\dagger, H\psi - \lambda G\psi \right\rangle_{E,\mathcal{D},\Omega} + \frac{\partial}{\partial \psi} \left[\hat{A}^\dagger \left(\frac{A}{V_R} \hat{\mathcal{P}} - \mathcal{P} \right) \right] \\ &= \left\langle H^\dagger \hat{\psi}^\dagger - \lambda G^\dagger \hat{\psi}^\dagger \right\rangle_{E,\mathcal{D},\Omega} + \hat{A}^\dagger \frac{A}{V_R} \left\langle \Sigma_E \right\rangle_{E,\mathcal{D},\Omega} \end{aligned} \quad (3.79)$$

$$\begin{aligned} \left\langle \frac{\partial R}{\partial \lambda} \right\rangle_{E,\mathcal{D},\Omega} &= \frac{\partial}{\partial \lambda} \left\langle \hat{\psi}^\dagger, H\psi - \lambda G\psi \right\rangle_{E,\mathcal{D},\Omega} + \frac{\partial}{\partial \lambda} \left[\hat{A}^\dagger \left(\frac{A}{V_R} \hat{\mathcal{P}} - \mathcal{P} \right) \right] \\ &= - \left\langle \hat{\psi}^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} \end{aligned} \quad (3.80)$$

$$\begin{aligned} \left\langle \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} &= \frac{\partial}{\partial A} \left\langle \hat{\psi}^\dagger, H\psi - \lambda G\psi \right\rangle_{E,\mathcal{D},\Omega} + \frac{\partial}{\partial A} \left[\hat{A}^\dagger \left(\frac{A}{V_R} \hat{\mathcal{P}} - \mathcal{P} \right) \right] \\ &= \frac{\hat{A}^\dagger}{V_R} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} \end{aligned} \quad (3.81)$$

The first step is to write an expression for \hat{A}^\dagger using Eq. (3.81):

$$\hat{A}^\dagger = \frac{V_R}{\hat{\mathcal{P}}} \left\langle \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega}. \quad (3.82)$$

Using Eq. (3.82) to substitute for \hat{A}^\dagger in Eq. (3.79), we find the following transport equation that must be solved for $\hat{\psi}^\dagger$

$$H^\dagger \hat{\psi}^\dagger - \lambda G^\dagger \hat{\psi}^\dagger = \frac{\partial R}{\partial \psi} - \frac{\left\langle A \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega}}{\left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega}} \Sigma_E, \quad (3.83)$$

subject to $\left\langle \hat{\psi}^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} = -\left\langle \frac{\partial R}{\partial \lambda} \right\rangle_{E,\mathcal{D},\Omega}$. Again, for $\hat{\psi}^\dagger$ to be unique, we require that

$$\left\langle \frac{\partial R}{\partial \psi} - \frac{\left\langle A \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega}}{\left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega}} \Sigma_E, \psi \right\rangle_{E,\mathcal{D},\Omega} = 0,$$

or that

$$\left\langle \frac{\partial R}{\partial \psi}, \psi \right\rangle_{E,\mathcal{D},\Omega} = \left\langle A \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega},$$

which is the same constraint on the form of the QOI as before. Given solutions to $\hat{\psi}^\dagger$ and \hat{A}^\dagger , our terminal condition for $N^\dagger(t_f)$ is

$$N^\dagger(t_f) = \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \hat{\psi}^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} - \hat{A}^\dagger \frac{A}{V} \frac{\partial \hat{\mathcal{P}}}{\partial N}. \quad (3.84)$$

If we solve Eqs. (3.75)–(3.78) for the adjoint variables, use Eqs. (3.79)–(3.81) to compute $\hat{\psi}^\dagger$ and \hat{A}^\dagger , and specify the terminal condition for N^\dagger with Eq. (3.84), we can use the following expression to compute the sensitivity of Q_f with respect to the

parameters:

$$\begin{aligned}
\frac{dQ_f}{dp} = & \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0} + \left[\left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \right]_{t=t_f} \\
& - \left[\left\langle \hat{\psi}^\dagger, \frac{\partial}{\partial p} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} + \hat{A}^\dagger \frac{\partial}{\partial p} \left[\frac{A}{V_R} \hat{\mathcal{P}} - \mathcal{P} \right] \right]_{t=t_f} \\
& + \int_{t_0}^{t_f} \left\{ \left\langle N^\dagger, \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial H\psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} + \lambda \left\langle \psi^\dagger, \frac{\partial G\psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \right. \\
& \quad \left. - A^\dagger \frac{A}{V_R} \frac{\partial \hat{\mathcal{P}}}{\partial p} + A^\dagger \frac{\partial \mathcal{P}(t)}{\partial p} \right\} dt. \tag{3.85}
\end{aligned}$$

The adjoint-based global QOI error estimate is also available for the k-eigenvalue case. Suppose we have estimated local truncation errors, r_N , r_ψ , and r_A during the forward solve, as well as errors in the initial conditions of N , r_{N_0} . Equation (3.34) gives the global QOI error estimate as

$$\begin{aligned}
\Delta Q_f = & - \int_{t_0}^{t_f} \left\{ N^\dagger \cdot r_N + \left\langle \psi^\dagger, r_\psi \right\rangle_{E,\mathcal{D},\Omega} + A^\dagger r_A \right\} dt \\
& + N^\dagger(t_0) \cdot r_{N_0} - \left\langle r_\psi, \hat{\psi}^\dagger \right\rangle_{E,\mathcal{D},\Omega} - r_A \hat{A}^\dagger. \tag{3.86}
\end{aligned}$$

3.4 Example of Extensibility: Addition of Heat Transfer Physics

In Sec. 3.1, we developed a framework for deriving adjoint equations in terms of general forward operators, parameters, and unknowns. We claim that this derivation affords flexibility and extensibility in a multiphysics modeling environment. In Secs. 3.2 and 3.3, we exercised the flexibility of the framework by applying our equations to derive adjoint equations for the source-driven and k-eigenvalue forms of the depletion equations, respectively. In this section, we illustrate the extensibility of the framework by adding heat transfer physics to the k-eigenvalue form of the depletion equations and deriving the form of the associated adjoint equations.

Suppose we model heat transfer using an energy balance equation with temperature, T , as the fundamental unknown. We write the equation as an algebraic constraint, namely

$$K(T)T = S(T),$$

where operator $K(T)$ is an energy loss operator (e.g. conduction, convection) and $S(T)$ is an energy production operator. This equation may be nonlinear in T if, for example, the thermal conductivity of the material depends on the material temperature. In the context of nuclide depletion calculations, the operator K will depend on the nuclide densities N , and the dominant energy source term will be fission, making S depend on both N and ψ . Likewise, microscopic cross-sections will vary with material temperature, and thus the Bateman and neutron transport operators will have a T dependence. We note that the energy balance equation could be written in time-differential form; we choose, however, to write it as an algebraic constraint because its time-dependent behavior in quasi-steady state reactor operation is closely related to that of the neutron flux. Finally, for simplicity in this example, we assume fixed-temperature boundary conditions for the heat transfer equation.

Appending the energy balance equation to Eqs. (3.54)–(3.57) and noting all dependencies (except for t , which we omit for brevity, but it does exist for each operator), we write

$$\frac{dN}{dt} - B(\psi, A, T, p)N = 0, \quad (3.87)$$

$$H(N, T, p)\psi - \lambda G(N, T, p)\psi = 0, \quad (3.88)$$

$$AP(N, T, p)\psi - \mathcal{P}(t) = 0 \quad (3.89)$$

$$K(N, T, p)T - S(N, A, \psi, T, p) = 0 \quad (3.90)$$

$$N(t = 0) = N_0(p), \quad t \in [t_o, t_f]. \quad (3.91)$$

As in Sec. 3.3, we begin by writing the abstraction of these governing equations in terms of our framework:

$$\begin{aligned} x^d &\leftarrow N \\ x^a &\leftarrow \begin{bmatrix} \psi/\lambda \\ A \\ T \end{bmatrix} \\ \lambda^d &\leftarrow N^\dagger \end{aligned}$$

$$\begin{aligned}
\lambda^a &\leftarrow \begin{bmatrix} \psi^\dagger \\ A^\dagger \\ T^\dagger \end{bmatrix} \\
F^d &\leftarrow \frac{dN}{dt} - BN \\
F^a &\leftarrow \begin{bmatrix} H\psi - \lambda G\psi \\ AP\psi - \mathcal{P}(t) \\ KT - S \end{bmatrix}.
\end{aligned}$$

We write our time-dependent QOI that depends on temperature as

$$Q = \int_{t_0}^{t_f} \left\langle R(N, \psi, A, T, p, t) \right\rangle_{E, \mathcal{D}, \Omega} dt. \quad (3.92)$$

The next step is to translate the general expressions Eqs. (3.7) and (3.8). We performed this for the k-eigenvalue equations in Sec. 3.3, which resulted first in Eqs. (3.61)–(3.64) and ultimately the k-eigenvalue depletion perturbation equations. We begin by showing the terms that should be added to the right-hand side of these equations.

First, in the adjoint Bateman equation, Eq. (3.61), we add the following term:

$$\frac{\partial}{\partial N} \left\langle T^\dagger, KT - S \right\rangle_{E, \mathcal{D}, \Omega} = \left\langle T^\dagger, \frac{\partial}{\partial N} [KT - S] \right\rangle_{E, \mathcal{D}, \Omega}.$$

In Eq. (3.62), we add

$$\frac{\partial}{\partial \psi} \left\langle T^\dagger, KT - S \right\rangle_{E, \mathcal{D}, \Omega} = - \left\langle T^\dagger, \frac{\partial}{\partial \psi} S \right\rangle_{E, \mathcal{D}, \Omega}.$$

No change will result in Eq. (3.63) because $\frac{\partial}{\partial \lambda} \left\langle T^\dagger, KT - S \right\rangle_{E, \mathcal{D}, \Omega} = 0$. Finally, for

Eq. (3.64), we would add

$$\frac{\partial}{\partial A} \left\langle T^\dagger, KT - S \right\rangle_{E,\mathcal{D},\Omega} = - \left\langle T^\dagger, \frac{\partial}{\partial A} S \right\rangle_{E,\mathcal{D},\Omega}.$$

The last step is to derive the form of the adjoint heat transfer equation. To do this, we will need to define operators that are analogous to b for the Bateman operator. Let these operators be

$$\left\langle k(T)T \right\rangle_{E,\mathcal{D},\Omega} = K(T)T$$

and

$$\left\langle s \right\rangle_{E,\mathcal{D},\Omega} = S.$$

Operators k and s will, of course, depend on the particular equation being employed by the modeler, a level of detail we do not discuss in this dissertation. Suppose that we can define an operator k^\dagger such that

$$\left\langle T^\dagger, k(T)T \right\rangle_{E,\mathcal{D},\Omega} = \left\langle k^\dagger(T)T^\dagger, T \right\rangle_{E,\mathcal{D},\Omega}$$

and

$$\left\langle k^\dagger(T)T \right\rangle_{E,\mathcal{D},\Omega} = K^\dagger(T)T.$$

If the operator K does indeed depend on T , the operator $k^\dagger(T)$ is not guaranteed to exist in general. The literature indicates that under certain smoothness and locality assumptions, this function does exist (in fact, multiple such operators may exist)[51, 30, 31]. We could also revert to the formulation by Cao and Petzold[19], which employs a linearization about a discretized set of forward equations and the vector dot product inner product to form the adjoint equations. Moving forward, we assume that K^\dagger does exist and satisfies the above expression.

The translation of Eq. (3.8) for the heat transfer equation is

$$0 = \frac{\partial}{\partial T} \left\langle N^\dagger, \frac{dN}{dt} - BN \right\rangle_{E,\mathcal{D},\Omega} + \frac{\partial}{\partial T} \left\langle \psi^\dagger, H\psi - \lambda G\psi \right\rangle_{E,\mathcal{D},\Omega} \\ + \frac{\partial}{\partial T} \left\langle A^\dagger, AP\psi - \mathcal{P} \right\rangle_{E,\mathcal{D},\Omega} + \frac{\partial}{\partial T} \left\langle T^\dagger, kT - s \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\partial R}{\partial T} \right\rangle_{E,\mathcal{D},\Omega}.$$

We perform similar term-by-term manipulations as those given for the adjoint transport equation in Sec. 3.3:

$$\begin{aligned} \frac{\partial}{\partial T} \left\langle N^\dagger, \frac{dN}{dt} - BN \right\rangle_{E,\mathcal{D},\Omega} &= -\frac{\partial}{\partial T} \left[N^\dagger \left(\frac{dN}{dt} - BN \right) \right] = -N^\dagger \frac{\partial BN}{\partial T} \\ \frac{\partial}{\partial T} \left\langle \psi^\dagger, H\psi - \lambda G\psi \right\rangle_{E,\mathcal{D},\Omega} &= \left\langle \psi^\dagger, \frac{\partial}{\partial T} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \\ \frac{\partial}{\partial T} \left\langle A^\dagger, AP\psi - \mathcal{P} \right\rangle_{E,\mathcal{D},\Omega} &= \left\langle A^\dagger, \frac{\partial}{\partial T} AP\psi \right\rangle_{E,\mathcal{D},\Omega} \\ \frac{\partial}{\partial T} \left\langle T^\dagger, kT - s \right\rangle_{E,\mathcal{D},\Omega} &= \frac{\partial}{\partial T} \left\langle T^\dagger, kT \right\rangle_{E,\mathcal{D},\Omega} - \frac{\partial}{\partial T} \left\langle T^\dagger, s \right\rangle_{E,\mathcal{D},\Omega} \\ &= \left\langle k^\dagger T^\dagger \right\rangle_{E,\mathcal{D},\Omega} - \left\langle T^\dagger, \frac{\partial s}{\partial T} \right\rangle_{E,\mathcal{D},\Omega} \end{aligned}$$

The resulting adjoint heat transfer equation would have the following form:

$$\begin{aligned} K^\dagger T^\dagger &= T^\dagger \frac{\partial S}{\partial T} + \left\langle \frac{\partial R}{\partial T} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A^\dagger, \frac{\partial}{\partial T} AP\psi \right\rangle_{E,\mathcal{D},\Omega} \\ &\quad - \left\langle \psi^\dagger, \frac{\partial}{\partial T} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} + N^\dagger \frac{\partial BN}{\partial T}. \end{aligned}$$

This completes our derivation of the adjoint depletion equations with heat transfer,

which are

$$\begin{aligned}
\frac{dN^\dagger}{dt} &= -B^\dagger N^\dagger + \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} + \left\langle T^\dagger, \frac{\partial}{\partial N} [KT - S] \right\rangle_{E,\mathcal{D},\Omega} \\
&\quad + \left\langle A^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} \\
H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger &= \frac{\partial R}{\partial \psi} + N^\dagger \frac{\partial bN}{\partial \psi} - A^\dagger \frac{A}{V_R} \Sigma_E + T^\dagger \frac{\partial S}{\partial \psi} \\
\left\langle \psi^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} &= 0 \\
A^\dagger &= \frac{V_R}{\hat{\mathcal{P}}} \left[\left\langle \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle N^\dagger \frac{\partial bN}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle T^\dagger, \frac{\partial S}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} \right] \\
K^\dagger T^\dagger &= T^\dagger \frac{\partial S}{\partial T} + \left\langle \frac{\partial R}{\partial T} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A^\dagger, \frac{\partial}{\partial T} AP\psi \right\rangle_{E,\mathcal{D},\Omega} \\
&\quad - \left\langle \psi^\dagger, \frac{\partial}{\partial T} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} + N^\dagger \frac{\partial bN}{\partial T} \\
N^\dagger(t_f) &= 0.
\end{aligned}$$

The corresponding sensitivity equation will have an additional term to account for the temperature dependence. Adding this term to Eq. (3.70), we write

$$\begin{aligned}
\frac{dQ}{dp} &= \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0} + \int_{t_0}^{t_f} \left\{ \left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right. \\
&\quad + \left\langle N^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} + A^\dagger \frac{\partial \mathcal{P}(t)}{\partial p} \\
&\quad \left. - \left\langle T^\dagger, \frac{\partial}{\partial p} [KT - S] \right\rangle_{E,\mathcal{D},\Omega} \right\} dt
\end{aligned}$$

This derivation is not an exact prescription, as this will depend on the particular form of the energy balance expression. Instead, the main takeaway is that the framework is readily extensible to include new physics and their coupling to the existing physics. For the modeler, it is only a matter of applying the provided abstraction to the particular forward equations.

3.5 Technique for Integrating the Sensitivity Equation

In the previous two sections, we derived sensitivity expressions (that is, Eqs. (3.46), (3.52), (3.70), and (3.85)) that require a time-integral of a cross-correlation of the forward and adjoint solutions. Here we describe our technique for evaluating these expressions.

Our strategy is to integrate the sensitivity equation along with the adjoint equations during the backwards time march. This approach offers two major advantages:

1. The sensitivity equation can be integrated using the built-in Runge-Kutta framework, resulting in an order of accuracy that is the same as the integration of the forward and adjoint equations.
2. The forward and adjoint solutions are already available during the adjoint solve.

An alternative approach is to store the forward and adjoint equations (or specific terms in the sensitivity equations) at quadrature points during the adjoint solve and then evaluate the time integral in a post-processing step. This could require a large amount of storage (or file I/O) and requires the extra work of interpolating the solutions to the quadrature points; further, it is likely to be less accurate than our technique.

Formally, we treat the integration as a series of appended dummy variables to the adjoint system. Consider a representative sensitivity equation, written for a particular parameter p :

$$\frac{dQ}{dp} = S_T^{(p)} - \int_{t_0}^{t_f} F^{(p)}(t)dt$$

where $S_T^{(p)}$ represents terms in the sensitivity equation that are evaluated at either $t = t_0$ or $t = t_f$, and $F^{(p)}(t)$ represents the cross-correlation terms. For each parameter

p , we append a variable $z^{(p)}(t)$ that satisfies

$$\begin{aligned}\frac{dz^{(p)}}{dt} &= F^{(p)}(t) \\ z^{(p)}(t_f) &= S_T^{(p)}.\end{aligned}$$

It is straightforward to show that the solution to this ODE at $t = t_0$ is indeed $\frac{dQ}{dp}$; that is

$$\frac{dQ}{dp} = z^{(p)}(t_0).$$

The ODE for z is integrated backwards in time using the same rule for integrating the adjoint equations.

3.6 Adjoint Equations for Three Particular QOIs and Non-Conforming QOIs

In the following three subsections, we discuss three QOIs that will be used throughout the remainder of this dissertation. They are a terminal nuclide inventory, a terminal total reaction rate, and the terminal reactivity. We conclude the section with a brief discussion of other possible QOI formulations and how these could be handled with the framework.

3.6.1 The terminal inventory QOI

The terminal inventory QOI is simply a count of the amount of a particular nuclide remaining in the reactor (or a sub-volume of the reactor) at $t = t_f$. We choose to express this QOI in units of [mol]. Let the function $\delta(r)$ be defined as

$$\delta(r) = \begin{cases} 0 & r \text{ not in QOI sub-volume} \\ 1 & r \text{ in QOI sub-volume} \end{cases}.$$

Therefore, the functional R appearing in either Eq. (3.47) or Eq. (3.74) has the form

$$R = \delta(r) \frac{1}{\langle \rangle_{E,\Omega}} \frac{N_k}{N_A},$$

where N_k is the nuclide density of the component of interest, and N_A is Avogadro's number. After integration over phase space, this QOI will have units of [mol].

From the standpoint of the formalism that we have developed, this QOI is straightforward to apply. In the source-driven case, the solution $\hat{\psi}^\dagger$ to Eq. (3.50), is zero. Thus, by Eq. (3.51), the adjoint terminal condition in those spatial cells where N_k is being inventoried is

$$N_k^\dagger(t_f) = \left\langle \frac{\partial R}{\partial N_k} \right\rangle_{E,\mathcal{D},\Omega} = \delta(r) \frac{V_c}{N_A},$$

where V_c is the cell volume. Not for all other nuclides, $N_j^\dagger(t_f) = 0$, $j \neq k$. The sensitivity equation, Eq. (3.52), is

$$\frac{dQ_f}{dp} = \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0} - \int_{t_0}^{t_f} \left\{ \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - G\psi - S_0] \right\rangle_{E,\mathcal{D},\Omega} - \left\langle N^\dagger, \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \right\}.$$

The terminal condition for this QOI in the k-eigenvalue case is a bit more complicated. First, by Eq. (3.82), we have that $\hat{A}^\dagger = 0$ because $\frac{\partial R}{\partial A} = 0$. Thus, the transport equation for the terminal condition vector $\hat{\psi}^\dagger$ is

$$H^\dagger \hat{\psi}^\dagger - \lambda G^\dagger \hat{\psi}^\dagger = 0,$$

which is satisfied by both the zero vector and the fundamental mode adjoint flux. We also found that the solution to this equation must satisfy $\left\langle \hat{\psi}^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} = -\left\langle \frac{\partial R}{\partial \lambda} \right\rangle_{E,\mathcal{D},\Omega} = 0$. A later analysis will show that the forward and adjoint fluxes are not orthogonal in this sense; therefore, $\hat{\psi}^\dagger = 0$ for this QOI. By Eq. (3.84), the terminal condition for N_k^\dagger is the same as above (and zero for all other nuclides). The sensitivity equation, Eq. (3.85) is then

$$\begin{aligned} \frac{dQ_f}{dp} &= \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0} - \int_{t_0}^{t_f} \left\{ \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} - \left\langle N^\dagger, \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \right. \\ &\quad \left. + A^\dagger \frac{A}{V_R} \frac{\partial}{\partial p} \left\langle \psi \Sigma_E \right\rangle_{E,\mathcal{D},\Omega} - A^\dagger \frac{\partial \mathcal{P}(t)}{\partial p} \right\} \end{aligned}$$

3.6.2 The terminal reaction rate QOI

The terminal reaction rate QOI is defined as the total reaction rate with a particular nuclide in a defined sub-volume of the reactor at $t = t_f$. It has units of $[\frac{1}{\text{sec}}]$. Let N_k be the density of the particular nuclide and σ be its total microscopic cross

section. The functional R appearing in the source driven case, Eq. (3.47), is

$$R = \delta(r)\psi\sigma N_k.$$

Then, the adjoint transport equation to be solved for $\hat{\psi}^\dagger$ at $t = t_f$ is

$$H^\dagger\psi^\dagger - G^\dagger\psi^\dagger = \frac{\partial R}{\partial\psi} = \delta(r)\sigma N_k.$$

It is interesting to note that this is exactly the adjoint transport equation that is required for performing steady-state adjoint transport calculations[20]. Given $\hat{\psi}^\dagger$, the source-driven terminal condition, Eq. (3.51), is

$$N_k^\dagger(t_f) = \left\langle \delta(r)\psi\sigma \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \hat{\psi}^\dagger, \frac{\partial}{\partial N_k} [H\psi - G\psi] \right\rangle_{E,\mathcal{D},\Omega}$$

for the nuclide in which the reaction is occurring and just

$$N_j^\dagger(t_f) = - \left\langle \hat{\psi}^\dagger, \frac{\partial}{\partial N_j} [H\psi - G\psi] \right\rangle_{E,\mathcal{D},\Omega}$$

for all other nuclides, $j \neq k$. Note: the adjoint terminal condition is non-zero, even for those nuclides that do not contribute directly to the QOI. Finally, the term containing $\frac{\partial R}{\partial p}$ in the sensitivity equation, Eq. (3.52), is non-zero if p represents σ . In this case, $\left\langle \frac{\partial R}{\partial \sigma} \right\rangle_{E,\mathcal{D},\Omega} = \left\langle \delta(r)\psi N_k \right\rangle_{E,\mathcal{D},\Omega}$.

The k-eigenvalue case for this QOI is as follows. First, the functional R must also carry the normalization constant A :

$$R = A\delta(r)\psi\sigma N_k.$$

By Eq. (3.82),

$$\hat{A}^\dagger = \frac{V_R}{\langle \psi \Sigma_E \rangle_{E,\mathcal{D},\Omega}} \left\langle \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} = \frac{V_R}{\langle \psi \Sigma_E \rangle_{E,\mathcal{D},\Omega}} \left\langle \psi \sigma N_k \right\rangle_{E,\mathcal{D},\Omega}.$$

We use this to substitute into Eq. (3.83) to form the adjoint transport equation for $\hat{\psi}^\dagger$:

$$\begin{aligned} H^\dagger \hat{\psi}^\dagger - \lambda G^\dagger \hat{\psi}^\dagger &= \frac{\partial R}{\partial \psi} - \hat{A}^\dagger \frac{A}{V_R} \Sigma_E \\ &= A \sigma N_k - \frac{\left\langle A \psi \sigma N_k \right\rangle_{E,\mathcal{D},\Omega}}{\langle \psi \Sigma_E \rangle_{E,\mathcal{D},\Omega}} \Sigma_E \end{aligned}$$

We've noted that a unique solution to this expression requires that the right hand side be orthogonal to $\psi(t_f)$. To see that this condition holds, write

$$\begin{aligned} &\left\langle \psi, A \sigma N_k - \frac{\left\langle A \psi \sigma N_k \right\rangle_{E,\mathcal{D},\Omega}}{\langle \psi \Sigma_E \rangle_{E,\mathcal{D},\Omega}} \Sigma_E \right\rangle_{E,\mathcal{D},\Omega} \\ &= \left\langle A \psi \sigma N_k \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\left\langle A \psi \sigma N_k \right\rangle_{E,\mathcal{D},\Omega}}{\langle \psi \Sigma_E \rangle_{E,\mathcal{D},\Omega}} \Sigma_E, \psi \right\rangle_{E,\mathcal{D},\Omega} \\ &= \left\langle A \psi \sigma N_k \right\rangle_{E,\mathcal{D},\Omega} - \frac{\left\langle A \psi \sigma N_k \right\rangle_{E,\mathcal{D},\Omega}}{\langle \psi \Sigma_E \rangle_{E,\mathcal{D},\Omega}} \left\langle \Sigma_E, \psi \right\rangle_{E,\mathcal{D},\Omega} \\ &= 0 \end{aligned}$$

The variables $\hat{\psi}^\dagger$ and \hat{A}^\dagger feed into the terminal condition expression, Eq. (3.84), and ultimately the sensitivity equation. Again, we note that the term $\frac{\partial R}{\partial p}$ is non-zero if σ is a member of p .

3.6.3 The terminal reactivity QOI

The last QOI is the reactivity, $\rho = \frac{k_{\text{eff}} - 1}{k_{\text{eff}}}$, at $t = t_f$. This QOI is only valid for k-eigenvalue problems. The functional R is trivial:

$$R = \frac{1}{\langle \rangle_{E,\mathcal{D},\Omega}} \frac{k_{\text{eff}} - 1}{k_{\text{eff}}}.$$

Therefore, $\frac{\partial R}{\partial A} = 0$, and by Eq. (3.82), $\hat{A}^\dagger = 0$. Similarly, $\frac{\partial R}{\partial \psi} = 0$, resulting in the adjoint transport equation for $\hat{\psi}^\dagger$:

$$H^\dagger \hat{\psi}^\dagger - \lambda G^\dagger \hat{\psi}^\dagger = 0$$

subject to

$$\langle \hat{\psi}^\dagger, G\psi \rangle_{E,\mathcal{D},\Omega} = -\left\langle \frac{\partial R}{\partial \lambda} \right\rangle_{E,\mathcal{D},\Omega} = -\left\langle \frac{\partial R}{\partial k_{\text{eff}}} \frac{\partial k_{\text{eff}}}{\partial \lambda} \right\rangle_{E,\mathcal{D},\Omega} = 1.$$

This constraint is simply a normalization on $\hat{\psi}^\dagger$. From here forward, let $\hat{\psi}^\dagger$ be the adjoint fundamental eigenfunction that satisfies this normalization. That is, let v^\dagger be the fundamental adjoint eigenfunction with arbitrary magnitude, and set

$$\hat{\psi}^\dagger = \frac{v^\dagger}{\langle v^\dagger, G\psi \rangle_{E,\mathcal{D},\Omega}}.$$

The adjoint terminal condition is then

$$N^\dagger(t_f) = -\left\langle \hat{\psi}^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega},$$

where all terms on the right-hand-side are evaluated at $t = t_f$. Again we see an interesting analog to a steady-state adjoint problem. This terminal condition is equivalent to the expression that gives the sensitivity of a steady-state QOI to the nuclide densities, N [20]. Finally, the sensitivity equation for the depletion perturbation problem with terminal reactivity QOI is

$$\begin{aligned} \frac{dQ}{dp} = & \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0} - \left[\left\langle \hat{\psi}^\dagger, \frac{\partial}{\partial p} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right]_{t=t_f} \\ & + \int_{t_0}^{t_f} \left\{ \left\langle N^\dagger, \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial H\psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} + \lambda \left\langle \psi^\dagger, \frac{\partial G\psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \right. \\ & \quad \left. - A^\dagger \frac{A}{V_R} \frac{\partial}{\partial p} \left\langle \psi \Sigma_E \right\rangle_{E,\mathcal{D},\Omega} + A^\dagger \frac{\partial \mathcal{P}(t)}{\partial p} \right\} dt. \end{aligned}$$

3.6.4 Non-conforming QOIs

Our framework assumed that the QOI may be written as either a time-integrated phase space integral over the forward solution or as a phase space integral of the forward solution evaluated only at the end of the simulation. Other interesting QOI formulations certainly exist. Here, we consider the case of a ratio of phase space integrals and the case of a QOI that is taken to be the maximum of some quantity, such as a peak fission rate.

First, consider a QOI of the form

$$Q = \int \frac{\left\langle R_1(x, p, t) \right\rangle}{\left\langle R_2(x, p, t) \right\rangle} dt,$$

as opposed to the form of Eq. (3.3). This could, for example, be a time-integrated total neutron production rate divided by a total neutron absorption rate in a particular material in a particular region of the reactor. We could still apply our framework

directly to a QOI of this form. We would just need to carry the proper differentiation through the steps that lead to the adjoint equations. On a related note, recall that the Fredholm alternative theorem is automatically satisfied if the form of the QOI satisfies Eq. (3.73). It is straightforward to show that if R_1 and R_2 are each bilinear in A and ψ , then the Fredholm alternative is also satisfied. Thus, as long as the QOI ratio is bounded, this particular form of the QOI does not present any challenges to our framework.

Second, consider a QOI that may be important for design optimization or safety considerations:

$$Q = \max_{V_i \in V} \int \left\langle R(x, p, t) \right\rangle_{V_i} dt.$$

For example, this could be the maximum pin-wise fission rate in the reactor. To apply the framework in this case, we envision an intermediate step. First, we would solve for the forward solution as usual. The extra step is then to search the phase space for the maximum (in essence, compute the QOI), which then implicitly defines a function R_i that is to be integrated to compute that QOI (it will be a step function that is non-zero only where the maximum occurs). Then, we would proceed with the adjoint calculation using R_i as the QOI functional, which would ultimately give us the sensitivity and/or error estimates for that particular maximum. If such a strategy was taken in an optimization loop, the intermediate step would be required to find the maximum and adjust R_i at each iteration.

4. STRATEGIES AND IMPLEMENTATION FOR MASSIVELY PARALLEL ARCHITECTURES

In this section, we describe the solvers that are implemented in the PDT code for the source-driven and k-eigenvalue formulations of the depletion perturbation equations. We begin the section with an overview of the PDT code and a broad description of the depletion solvers, adjoint transport solver, and the adjoint depletion solver. These descriptions are meant to be high-level and are supplemented by the detailed descriptions in Appendix B.

The second part of the section describes the family of checkpointing algorithms that we have developed, implemented, and tested in PDT. Three of these algorithms take advantage of the dimensionality reduction that occurs in the source iteration procedure, and the result is a decreased memory and I/O load on the host machine. We analyze the schemes in terms of their FLOP costs, memory costs, and I/O costs. Our hypothesis that the new schemes will improve time to solution for large problems on large processor counts will be tested in a later section.

4.1 Description of the PDT Software: A Massively Parallel Discrete Ordinates Transport Solver

In this section we give a broad overview of PDT, a software project designed to solve large scale discrete ordinates transport problems. The project is a collaboration of graduate students, research staff members, and faculty members of the nuclear engineering, computer science, and mathematics departments at Texas A&M University. The code is written in the C++ programming language and includes solvers for neutronics, radiative transfer, and charged-particle transport problems. The computational strategy is focused on the efficient solution of large scale problems at massively parallel levels of computing. Recently published results[1] illustrate a provably optimal solution strategy and leading-edge scaling results on $\mathcal{O}(100,000)$ cores.

For this dissertation, we are focused on the family of neutronics solvers in PDT. A neutronics problem is defined by the user via an input file. This file specifies the type of neutronics problem to be solved, discretization schemes, iterative algorithm, and the materials and geometry that compose the problem domain. Based on these settings, the main PDT driver file chooses the proper path through the code to compute and report the desired solution.

The first step is always to process the problem materials and geometry; this includes reading in material cross sections and densities, dividing the domain into spatial cells, and composing the homogeneous material mixtures that exist in each cell. In PDT, individual nuclides are referred to as components, and a mixture of components in a spatial cell is referred to as a material. The macroscopic cross sections of a material, which appear in the neutron transport equation, are a function of the microscopic cross sections, σ , and densities, N , of the material's components.

For example, the macroscopic cross section for interaction k and group g is

$$\Sigma_{k,g} = \sum_{i=1}^I N_i \sigma_{i,k,g},$$

where I is the number of components in the material.

Next the software chooses the appropriate solver for the particular problem. The solvers are written in a hierarchical design, beginning with the simple steady-state solver called NeutronicsSolver. A brief description of the solver hierarchy, roughly in order of increasing complexity, is as follows:

1. NeutronicsSolver: Solve the steady-state, fixed-source transport equation

$$\hat{H}\psi = S_0.$$

For k-eigenvalue problems (described next), $\hat{H} = H$ and S_0 is the latest update for the fission source. For time-dependent or true steady-state problems, $\hat{H} = H - G$ and S_0 is a prescribed, fixed neutron source. All other solvers derive from NeutronicsSolver and ultimately use repetitive calls to this solver to complete a calculation.

2. KSolver: Solve the k-eigenvalue formulation of the transport equation,

$$H\psi - \frac{1}{k_{\text{eff}}}G\psi = 0,$$

using the power iteration method. The power iteration method iterates on the fission source to compute the fundamental mode eigenvalue/eigenvector solution to this equation. Each iterate on the fission source requires a fixed-source transport solve. KSolver outputs the fundamental eigenvalue, k_{eff} , and funda-

mental mode flux solution, ψ , normalized by a user-defined power constraint.

3. SSAdjointSolver: Solve a steady-state adjoint transport problem of the form

$$H^\dagger \psi^\dagger - G^\dagger \psi^\dagger = S^\dagger.$$

The form of S^\dagger will depend on the particular quantity of interest. This solver, which includes a wrapper around NeutronicsSolver for solving the forward problem, will be described in more detail in Sec. 4.3.

4. BPSolver: Solve either the source-driven or k-eigenvalue form of the depletion equations. An operator-splitting implicit scheme and general Runge-Kutta half-implicit or half-explicit schemes are available. Both schemes are ultimately wrappers around NeutronicsSolver or KSolver. This solver will be described in more detail in Sec. 4.2.
5. MSASolver: Use the method of successive approximations to solve the adjoint, non-homogeneous, k-eigenvalue transport equation (e.g. (3.66)) subject to the orthogonality constraint Eq. (3.67). The solver requires that a solution to the homogeneous k-eigenvalue adjoint problem be available. This solver will be described in more detail in Sec. 4.4.
6. DplAdjointSolver: Solve the depletion perturbation equations. This solver is the main contribution of this dissertation. It includes a wrapper around BPSolver for solving and checkpointing the forward solution, as well as for formulating and solving the adjoint equations. This solver will be described in more detail in Sec. 4.5.

The core functionality of the neutronics solvers is the fixed-source transport solve.

As the list indicates, all of the derived solvers ultimately iteratively solve fixed-source transport problems (either inverting a forward or adjoint operator). Although we do not address it in this dissertation, we note that PDT also contains a family of solvers for the time-dependent neutron transport equation as well as solvers for thermal radiative transfer, electron transport, and gamma transport.

PDT employs an iterative algorithm known as source iteration to converge to a fixed-source transport solution. It also offers Krylov methods such as GMRES, Conjugate Gradient (CG), and BiCG-STAB. This dissertation does not require an in-depth description of these algorithms (see refs. [52, 53] for such detail), but a broad description will provide useful background for the subsequent sections.

Each iteration of the power iteration algorithm requires a “sweep” along each of the discrete ordinates, or angular directions, being treated in the problem. For a particular angle, a sweep begins at the inflow problem boundary, where the incident angular flux is specified by a boundary condition. For each cell on this boundary, the exiting angular flux is computed, which serves as a boundary condition for the cell’s “downstream” neighbors. Each sweep for each angle ultimately progresses from the upstream problem boundary to the downstream problem boundary.

The exact formulation of the exiting flux for a particular angle in a particular cell depends on the spatial discretization method. PDT offers a variety of discontinuous finite-element methods (DFEMs), which can handle arbitrary cell shapes, as well as weighted-diamond methods, which are restricted to orthogonal grids. There is long-characteristics implementation as well for arbitrary grids, but in its current form it requires significantly longer run times than the DFEMs. No matter the discretization scheme, however, the source iteration procedure uses the previous iterate’s flux solution to form the current iterate’s source terms. In other words, the angular flux is computed by inverting the streaming plus collision terms in the transport equation,

using the previous iterate to compute the scattering and fission source terms. The iterative procedure is summarized as

$$\Omega \cdot \nabla \psi^\ell + \Sigma_t \psi^\ell = S_S(\psi^{\ell-1}) + S_F(\psi^{\ell-1}) + S_0,$$

where ℓ is an iteration index and S_S and S_F represent the scattering and fission sources, respectively. This scheme is repeated until some measure of the difference from one iteration to the next is within a user-defined tolerance.

Source iteration has a simple physical interpretation. If the initial guess for the flux solution is $\psi^0 = 0$, then the first source iteration, ψ^1 , is the uncollided flux, or all neutrons that have been directly emitted by the fixed source S_0 , or have flowed in from the problem boundary but have not yet collided with a nucleus. The second iterate, ψ^2 , is the once-collided flux, or neutrons that have undergone one scattering or fission event. The third iterate is the twice-collided flux, and so on. Because all uncollided neutrons must ultimately die by absorption or leakage from the problem, this procedure is guaranteed to converge (so long as the system is subcritical). The convergence rate may be arbitrarily slow, however, for a problem where neutrons undergo many scattering and/or fission interactions before they disappear via absorption or leakage.

Krylov methods usually converge faster than source iteration with little additional cost per iteration. They choose the ℓ -th iteration to be the “best” member, in some defined sense, of a vector space known as a Krylov subspace. The main point here is that they rely on sweeps, just as source iteration does, and simply manipulate the sweep inputs and outputs to achieve their ends.

In summary, the family of neutronics solvers in PDT ultimately require the repeated solution of the fixed-source transport equation. PDT employs the source

iteration technique with global, boundary-to-boundary sweeps and optional Krylov acceleration to converge to a solution within a specified tolerance. After a fixed source solve, each spatial cell contains angular and/or scalar flux solutions, which are used by the various solvers to update the particular problem (e.g. update component densities) and formulate subsequent fixed-source problems. In the following sections, we describe how the depletion, adjoint, and depletion perturbation solvers interact with PDT’s fixed-source solver.

4.2 Description of the Depletion Solver in PDT

In this section, we describe the implementation of two algorithms for solving either the fixed-source or k-eigenvalue form of the depletion equations in PDT. This description is supplemented by detailed documentation in Appendix B. Recall from Sec. 2.1 that our model is an independent set of Bateman equations at each spatial point (that is, we don’t account for migration of nuclides). We also make the basic assumption that the microscopic cross sections are not functions of the nuclide number densities. This assumption neglects changes in spatial self-shielding that occur as the nuclide inventories change.

This first depletion algorithm, which we will call the operator splitting (OS) algorithm, uses a linear approximation for the flux magnitude over “flux time steps” and solves the Bateman equation over shorter “density time steps.” Thus, the OS algorithm has an implicit component. The second algorithm, which we will call the Runge-Kutta (RK) algorithms, solves the depletion equations according to a general Runge-Kutta time integration rule.

4.2.1 The operator splitting solution technique

This technique solves the quasi-static version of the depletion equations by assuming a linear shape of the flux magnitude over a series of time steps. Pseudocode

for advancement over a time step $\Delta t_\phi = t_{n+1} - t_n$ is given in Fig. 4.1. The scheme begins with a guess that the cell-averaged scalar flux in each group is constant over the time step. Using this guess, the densities are integrated over the time step to solve for the end-of-time-step densities. Then, a new guess for ϕ_{n+1} is computed using these densities. If the relative change in the guess for N_{n+1} and/or ϕ_{n+1} is small enough, the procedure is finished and we move on to the next time step.

- given: $\Phi(t_n), N(t_n);$
- guess $\Phi(t_{n+1})=\Phi(t_n), N(t_{n+1})=N(t_n)$, iter=0, converged=false;
- while (iter < maxIter) && (converged==false)
 1. iter++;
 2. let $\Phi(t_n+t)=\Phi(t_n)+(t-t_n)/(t_{n+1}-t_n)*(\Phi(t_{n+1})-\Phi(t_n));$
 3. solve for $N(t_{n+1});$
 4. solve for $\Phi(t_{n+1});$
 5. converged = check point-wise convergence;

Figure 4.1: Pseudocode for the operator splitting depletion algorithm

We consider some of these steps in more detail. In step 3, the Bateman solutions are integrated from t_n to t_{n+1} using the latest guess for the slope of flux magnitude over that time step. The time step for this integration, Δt_N , may be smaller than Δt_ϕ , which is advantageous for three reasons:

1. The nuclide densities tend to change on time scales much shorter than the flux solution, especially for those nuclides with short half lives;
2. The Bateman solution is much cheaper to obtain than the global flux solution because it is a simple ODE; and

3. The integration of the Bateman equation within a cell is totally independent of that in other cells, so each processor is free to deplete its nuclides in a completely parallel fashion.

We choose to use a general, implicit Runge-Kutta scheme to integrate the Bateman equation within each cell. This requires one linear solve of size I per Runge-Kutta stage, where I is the number of nuclides in the cell. We use a direct LU decomposition solver for these systems. Even for problems with ratio $\frac{\Delta t_\phi}{\Delta t_N} = \mathcal{O}(100)$, we find that the time spent on these linear systems is negligibly small compared to the time spent computing ϕ . The advantage of the implicit solver is for resolving rapidly changing component densities with relatively large time steps.

In step 5, the latest 2 guesses for ϕ_{n+1} and N_{n+1} are compared to determine whether the iterative scheme has converged or not. This is a point-wise convergence test, meaning that we are testing the relative change in these quantities at each point where they are defined in the problem. The user specifies whether to check ϕ , N , or both, as well as their respective tolerances, in the input file. The user also specifies the maximum number of iterations to perform over a given Δt_ϕ . These options allow for flexibility in the scheme. For example, the scheme can be made completely explicit by setting maxIter=1, which would correspond to a piece-wise constant guess for the flux magnitude. Similarly, if maxIter=2, the scheme could be interpreted as a predictor-corrector technique, where one flux update is allowed per time step.

The main advantages of this algorithm are first, that implicit flux information is used to integrate the Bateman equation, and second, that the Bateman equations can be integrated at a time resolution that is not limited by the frequency of the transport solves. Overall, these advantages help reduce time step limitations im-

posed by initial burn-ins and/or short lived components, allowing for more efficient simulation of realistic reactor operational cycles. For example, clever choices of Δt_ϕ , Δt_N , tolerances, and iteration limits early in a reactor cycle will help resolve shorter time-scale physics. After the reactor has reached a more quasi-equilibrium state, these settings can be relaxed to decrease time to solution.

4.2.2 The Runge-Kutta solution technique

This time marching technique employs a general-order, half-explicit or half-implicit Runge-Kutta scheme to integrate the depletion equations over a series of time steps of length Δt . The half-explicit and half-implicit schemes were outlined in Sec. 2.3 by Eqs. (2.10) and (2.11), respectively.

For both schemes, either the stage vector, N_i , or the Bateman derivative, $B_i N_i$, must be stored for each stage in order to compute subsequent stage vectors, including the final stage vector N_{n+1} . We choose to store $B_i N_i$ so that we do not have to recompute B and therefore do not have to store s flux stage vectors at each time step. Pseudocode for advancing the densities from t_n to t_{n+1} is given in Fig. 4.2.

```

• given:  $\Phi(t_n)$ ,  $N(t_n)$ ;
• for  $s=1:nStage$ 
    1. if ( $s > 1$ )
        • compute  $\Phi_s$ ;
    2. compute and store  $B_s N_s$ ;
    3. if ( $s < nStage$ )
        • compute  $N_{s+1}$ ;
• compute  $N(t_{n+1})$ ;
• compute  $\Phi(t_{n+1})$ ;
• if (embedded)
    • compute  $N^{p+1}(t_{n+1})$ ;
    • compute  $\Phi^{p+1}(t_{n+1})$ ;
    • compute error estimate;

```

Figure 4.2: Pseudocode for the Runge-Kutta depletion algorithm

This algorithm has the advantage of flexibility in choosing the Runge-Kutta scheme. Higher order schemes may be used when necessary to improve stability and/or accuracy in the solution. Perhaps more importantly, the sub-cycled half-explicit (Eq. (2.12)) and half-implicit (Eq. (2.13)) schemes allow for short Bateman time-steps relative to the overall depletion time steps, alleviating time step limitations. These sub-cycled schemes are limited to first order accuracy. Lastly, embedded Runge-Kutta schemes may be used to generate an order $p + 1$ accurate solution in addition to the order p solution, allowing for an end-of-time-step error calculation.

As described in Sec. 2.5.2, the adjoint of the Runge-Kutta scheme is well defined, allowing for a fairly straightforward adaptation of the forward solver to solve the adjoint problem. We choose to only use self-adjoint Runge-Kutta schemes. Most of the well-known schemes are self-adjoint (explicit and implicit Euler, modified Euler, RK4), and these are most straightforward to implement.

4.3 Description of the Adjoint Transport Solver in PDT

Here we describe the steady-state adjoint transport solver in PDT. The purpose of this solver is to produce an adjoint flux solution, $\psi^\dagger(r, E, \Omega_n)$ in addition to a forward flux solution, $\psi(r, E, \Omega)$, and cross-correlate them to produce sensitivity information about a particular quantity of interest. We introduced an example problem of this type in Sec. 2.4.1. Recall that we defined a QOI,

$$Q = \left\langle R(\psi, p) \right\rangle_{E, \mathcal{D}, \Omega},$$

derived from the solution ψ to the forward problem

$$H\psi - G\psi = S_0$$

with vacuum boundary condition. We then showed that the sensitivity of this QOI with respect to each parameter p can be computed as

$$\frac{dQ}{dp} = \left\langle \frac{\partial R}{\partial p} \right\rangle_{E, \mathcal{D}, \Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - G\psi - S_0] \right\rangle_{E, \mathcal{D}, \Omega},$$

where ψ^\dagger satisfies the adjoint equation

$$H^\dagger \psi^\dagger - G^\dagger \psi^\dagger = S^\dagger,$$

where $S^\dagger \equiv \frac{\partial R}{\partial \psi}$.

To write this solver, we first needed to implement a solver for the adjoint transport equation given a right-hand-side. In Sec. 2.4.3, we showed that the forward solver can be “tricked” into solving for $\psi^\dagger(r, E, -\Omega_n)$, the adjoint angular flux in the direction opposite of Ω_n , with a single modification: a transpose of the scattering and fission

source terms. This is a convenient result because we are not required to re-write (and therefore re-test and re-verify) an adjoint transport solver; we need only to transpose the energy transfer terms when we are in adjoint mode.

To accomplish this, we added three members to the code. First, we implemented classes that compute the transpose of the scattering and fission source matrices; these are simple adaptations of the existing source classes in the code. Second, we added a flag *adjoint_mode*, which tells the code whether or not to use the transposed versions of the source operators. This flag is manipulated as required by the solvers depending on which stage (forward or adjoint) the problem is in. Finally, the quadrature sets were modified such that each angle knows the index of its corresponding opposite. This way, anytime we are integrating over angle (e.g. to compute an inner product), the correct adjoint flux can be accessed quickly.

We implemented a family of classes that provide QOI information to the solver. A given QOI class can compute the actual quantity of interest, provide S^\dagger prior to the adjoint solve, and compute $\left\langle \frac{\partial R}{\partial p} \right\rangle_{E,D,\Omega}$. The QOI also handles the cross-correlation and, as we will describe in the next section, performs the time integration of the sensitivity equation in depletion perturbation problems.

Pseudocode for the steady-state adjoint transport solver is given in Fig. 4.3. Steps 1 and 2 are fairly straightforward. The forward flux is computed using NeutronicSSolver or KSolver, and the user-requested QOI is calculated and reported to an output file. In step 3, we write the solution to memory; that is, each spatial element stores a copy the converged forward angular flux vector. This will be used to compute $\frac{dQ}{dp}$ in step 6.

1. solve forward problem for ψ ;
2. compute QOI;
3. store ψ in RAM;
4. compute and set S^+ ;
5. solve adjoint problem for ψ^+ ;
6. cross-correlate ψ and ψ^+ to compute dQ/dp ;

Figure 4.3: Pseudocode for the steady-state adjoint solver

In step 4, the adjoint source S^\dagger is computed and stored on each spatial element.

Note that S^\dagger is related to the functional R but is not integrated over phase space. For example, if the QOI is the total reaction rate with nuclide k over the whole domain, the QOI expression is

$$Q = \left\langle \sigma_{t,k} \psi N_k \right\rangle_{E,\mathcal{D},\Omega}.$$

Therefore

$$S^\dagger = \sigma_{t,k} N_k.$$

In step 5, the *adjoint_mode* flag is turned on. In a fixed-source calculation, the NeutronicsSolver class iterates (with transposed scattering and fission energy transfer terms) until ψ^\dagger is converged. In a steady-state k-eigenvalue calculation, the MSA-Solver performs the method of successive approximation (see below), which includes a wrapper around NeutronicsSolver, and maintains the proper orthogonality relationships. For both cases, the result is a converged ψ^\dagger .

Finally, in step 6, the QOI class computes $\left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega}$ and the correct expression for $\frac{dQ}{dp}$ for each p specified by the user. These terms are described in detail in Appendix B; as a brief example, however, consider the sensitivity equation for our

reaction rate QOI and the parameter $\sigma_{t,g}$ for nuclide k :

$$\frac{dQ}{d\sigma_{t,g,k}} = \left\langle \psi_g N_k \right\rangle_{\mathcal{D},\Omega} - \left\langle \psi_g^\dagger, \psi_g N_k \right\rangle_{\mathcal{D},\Omega}.$$

Note that special care must be taken to use the correct ψ_g^\dagger with regard to angle in the second inner product.

4.4 Description of the Method of Successive Approximations Implementation in PDT

Here we describe our implementation of the method of successive approximations (MSA) in PDT. This solver is used to converge a solution to an equation of the form

$$H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger = S^\dagger \quad (4.1)$$

subject to the orthogonality constraints

$$\left\langle \psi, S^\dagger \right\rangle_{E,\mathcal{D},\Omega} = 0 \quad (4.2)$$

$$\left\langle \psi^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} = 0. \quad (4.3)$$

Equations and constraints of this form appear in the adjoint equations corresponding to the k-eigenvalue formulation of the forward depletion equations. We formulated our MSA scheme based on a technical report by Oblow[54]. We depart from his analysis, however, to show an improved convergence result, which is discussed in the following paragraphs.

We begin with some analysis that lends mathematical insight into the orthogonal-

ity conditions. Consider the forward and adjoint homogeneous eigenvalue equations:

$$Hv - \lambda Gv = 0$$

$$H^\dagger v^\dagger - \eta G^\dagger v^\dagger = 0,$$

which have eigenvalue/eigenvector solutions $\{\lambda_i, v_i\}_{i=0}^\infty$ and $\{\eta_i, v_i^\dagger\}_{i=0}^\infty$, respectively.

Let the index i be ordered such that λ_0 , the fundamental mode, is the smallest eigenvalue in magnitude. By this definition, $v_0 \equiv \psi$. We first claim and prove that v and v^\dagger are biorthogonal, meaning

$$\langle v_i, H^\dagger v_j^\dagger \rangle_{E,\mathcal{D},\Omega} = \langle v_i, \eta_j G^\dagger v_j^\dagger \rangle_{E,\mathcal{D},\Omega} = \begin{cases} 0 & i \neq j \\ \alpha_i & i = j \end{cases}.$$

First, we write the following inner products

$$\begin{aligned} \langle v_j^\dagger, Hv_i \rangle_{E,\mathcal{D},\Omega} - \langle v_j^\dagger, \lambda_i Gv_i \rangle_{E,\mathcal{D},\Omega} &= 0 \\ \langle v_i, H^\dagger v_j^\dagger \rangle_{E,\mathcal{D},\Omega} - \langle v_i, \eta_j G^\dagger v_j^\dagger \rangle_{E,\mathcal{D},\Omega} &= 0, \end{aligned}$$

and subtract the second line from the first to write

$$\langle v_j^\dagger, Hv_i \rangle_{E,\mathcal{D},\Omega} - \langle v_i, H^\dagger v_j^\dagger \rangle_{E,\mathcal{D},\Omega} - \langle v_j^\dagger, \lambda_i Gv_i \rangle_{E,\mathcal{D},\Omega} + \langle v_i, \eta_j G^\dagger v_j^\dagger \rangle_{E,\mathcal{D},\Omega} = 0.$$

Applying our inner product rules for operators H and G , we eventually find

$$(\eta_j - \lambda_i) \langle v_j^\dagger, Gv_i \rangle_{E,\mathcal{D},\Omega} = 0. \quad (4.4)$$

If $i \neq j$, the first term is non-zero, requiring that the second term be zero and proving

part of the claim. To see the case for $i = j$, re-write the eigenvalue equations as

$$\frac{1}{\lambda_i} v_i = (H^{-1}G)v_i \quad (4.5)$$

$$\frac{1}{\eta_j} v_j^\dagger = [H^\dagger]^{-1} G^\dagger v_j^\dagger \quad (4.6)$$

Similar manipulations lead to the equality

$$\lambda_i \langle v_j^\dagger, v_i \rangle_{E,\mathcal{D},\Omega} - \eta_j \langle v_i, v_j^\dagger \rangle_{E,\mathcal{D},\Omega} = (\lambda_i - \eta_j) \langle v_i, v_j^\dagger \rangle_{E,\mathcal{D},\Omega} = 0. \quad (4.7)$$

The analysis begins by expanding ψ^\dagger , the solution to the inhomogenous eigenvalue problem (the adjoint flux that we ultimately wish to compute) in terms of the homogeneous adjoint eigenvectors and the action of the adjoint transport operator on S^\dagger :

$$\psi^\dagger = [H^\dagger]^{-1} S^\dagger + \sum_{i=0}^{\infty} b_i v_i^\dagger.$$

This form differs from that of Oblow, and an explanation of this form is given below. Note that the operator $H^\dagger - \lambda_0 G^\dagger$ maps v_0^\dagger , as well as any scaling of that vector, $b_0 v_0^\dagger$, to zero. Therefore, in order for ψ^\dagger to be unique, this mode (the fundamental mode) must not be present in ψ^\dagger . In other words, we must enforce $b_0 = 0$. It turns out that a combination of the Fredholm alternative condition, Eq. (4.2), and the fission orthogonality condition, Eq. (4.3), works to filter out the fundamental mode. To see

this, we begin with a trivial expression which leads to the Fredholm condition:

$$\begin{aligned}
0 &= \left\langle \psi^\dagger, 0 \right\rangle_{E,\mathcal{D},\Omega} \\
&= \left\langle \psi^\dagger, H\psi - \lambda_0 G\psi \right\rangle_{E,\mathcal{D},\Omega} \\
&= \left\langle H^\dagger \psi^\dagger - \lambda_0 G^\dagger \psi^\dagger, \psi \right\rangle_{E,\mathcal{D},\Omega} \\
&= \left\langle S^\dagger, \psi \right\rangle_{E,\mathcal{D},\Omega}
\end{aligned}$$

If we now incorporate the fission orthogonality condition and substitute our expansion for ψ^\dagger , we find

$$\begin{aligned}
0 &= \left\langle \psi^\dagger, H\psi \right\rangle_{E,\mathcal{D},\Omega} - \lambda_0 \left\langle \psi^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} \\
&= \left\langle \psi^\dagger, H\psi \right\rangle_{E,\mathcal{D},\Omega} \\
&= \left\langle [H^\dagger]^{-1} S^\dagger, H\psi \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \sum_{j=0}^{\infty} b_j v_j^\dagger, H\psi \right\rangle_{E,\mathcal{D},\Omega} \\
&= \left\langle H^\dagger [H^\dagger]^{-1} S^\dagger, \psi \right\rangle_{E,\mathcal{D},\Omega} + b_0 \left\langle v_0^\dagger, H\psi \right\rangle_{E,\mathcal{D},\Omega} \\
&= \left\langle S^\dagger, \psi \right\rangle_{E,\mathcal{D},\Omega} + b_0 \alpha_0 \\
&= b_0 \alpha_0
\end{aligned}$$

which requires that $b_0 = 0$. Again, the orthogonality conditions filter fundamental mode contamination out of ψ^\dagger , allowing for a unique solution.

The convergence analysis also requires an expansion related to the adjoint source. Again, our analysis differs from that of Oblow, who proposed $S^\dagger = \sum_{j=0}^{\infty} a_j \lambda_j G^\dagger v_j^\dagger$. We find that $G^\dagger v^\dagger$ may not form a basis in general. For example, S^\dagger for a time-

dependent QOI is (see Eq. (3.66))

$$S^\dagger = \frac{\partial R}{\partial \psi} + \tilde{N}^\dagger \frac{\partial bN}{\partial \psi} - \tilde{A}^\dagger \frac{A}{V_R} \Sigma_E,$$

where R is the functional being phase-space integrated to form the QOI. If R is non-zero in a region with no fission (e.g. absorption in a control rod), then it cannot be represented by $G^\dagger v^\dagger$, which is zero by definition in a non-fissioning region. Thus, $G^\dagger v^\dagger$ is not a suitable basis in general.

Instead, we propose an expansion of the form

$$[H^\dagger]^{-1} G^\dagger [H^\dagger]^{-1} S^\dagger = \sum_{j=0}^{\infty} a_j v_j^\dagger.$$

This can be manipulated to write

$$\begin{aligned} G^\dagger [H^\dagger]^{-1} S^\dagger &= \sum_{j=0}^{\infty} a_j H^\dagger v_j^\dagger \\ &= \sum_{j=0}^{\infty} a_j \lambda_j G^\dagger v_j^\dagger, \end{aligned}$$

and thereby has some similarity to the Obloj expansion. It does not, however, have the obvious contradiction related to a projection to zero in non-fissioning regions. We do not offer a formal proof that this expansion holds in general. We do note, however, that the operator $G^\dagger [H^\dagger]^{-1}$ is compact (it is an integration over space, angle, and energy) and is therefore more likely to be well-behaved in terms of the basis. We proceed under the assumption that the expansion holds.

We first use a series of manipulations and the biorthogonality properties of the eigenmodes to write a relationship between the a_j 's and the b_j 's. We first operate

on the adjoint flux equation with $[H^\dagger]^{-1}G^\dagger[H^\dagger]^{-1}$ from the left as follows:

$$0 = H^\dagger\psi^\dagger - \lambda_0 G^\dagger\psi^\dagger - S^\dagger \\ = [H^\dagger]^{-1}G^\dagger\psi^\dagger - \lambda_0 [H^\dagger]^{-1}G^\dagger[H^\dagger]^{-1}G^\dagger\psi^\dagger - [H^\dagger]^{-1}G^\dagger[H^\dagger]^{-1}G^\dagger S^\dagger.$$

We next take the inner product of this operator with an arbitrary forward mode, v_i , and manipulate using our expansion for ψ^\dagger :

$$0 = \left\langle v_i, \sum_{j=0}^{\infty} b_j [H^\dagger]^{-1}G^\dagger v_j^\dagger + [H^\dagger]^{-1}G^\dagger[H^\dagger]^{-1}S^\dagger - \lambda_0 \sum_{j=0}^{\infty} b_j [H^\dagger]^{-1}G^\dagger[H^\dagger]^{-1}G^\dagger v_j^\dagger \right. \\ \left. - \lambda_0 [H^\dagger]^{-1}G^\dagger[H^\dagger]^{-1}G^\dagger[H^\dagger]^{-1}S^\dagger - [H^\dagger]^{-1}G^\dagger[H^\dagger]^{-1}S^\dagger \right\rangle_{E,\mathcal{D},\Omega} \\ = \left\langle v_i, \sum_{j=0}^{\infty} b_j \frac{1}{\lambda_j} v_j^\dagger - \lambda_0 \sum_{j=0}^{\infty} b_j \frac{1}{\lambda_j^2} v_j^\dagger - \lambda_0 \sum_{j=0}^{\infty} a_j [H^\dagger]^{-1}G^\dagger v_j^\dagger \right\rangle_{E,\mathcal{D},\Omega} \\ = \left\langle v_i, \sum_{j=0}^{\infty} b_j \frac{1}{\lambda_j} v_j^\dagger - \lambda_0 \sum_{j=0}^{\infty} b_j \frac{1}{\lambda_j^2} v_j^\dagger - \lambda_0 \sum_{j=0}^{\infty} a_j \frac{1}{\lambda_j} v_j^\dagger \right\rangle_{E,\mathcal{D},\Omega} \\ = \left\langle v_i, \left[\frac{b_i}{\lambda_i} - \frac{\lambda_0 b_i}{\lambda_i^2} - \frac{\lambda_0 a_i}{\lambda_i} \right] v_i^\dagger \right\rangle_{E,\mathcal{D},\Omega}$$

In order for this to hold, we must have

$$b_i \left[\frac{1}{\lambda_i} - \frac{\lambda_0}{\lambda_i^2} \right] = \frac{\lambda_0}{\lambda_i} a_i,$$

or after some manipulation,

$$b_i = \lambda_0 \left[\frac{1}{1 - \frac{\lambda_0}{\lambda_i}} \right] a_i.$$

The converse of this expression implies that if $b_0 = 0$, then $a_0 = 0$, indicating that the expansion for S^\dagger must also not contain any fundamental mode contamination. The effects of this contamination and our strategy for filtering it out of the solution

are described below.

We now analyze the convergence rate of the MSA scheme and show that it does indeed converge to the solution (ψ^\dagger) to Eq. (4.1) in the limit of infinite iterations. The scheme is similar to the power iteration method in that the (adjoint) fission source is iteratively updated until convergence. The value of λ_0 , however, is fixed: it is the eigenvalue computed during the forward solve. So, MSA is not formally a filtering scheme; instead, we are simply creeping up on the solution. The iterative scheme is summarized as follows:

$$H^\dagger \psi^{\dagger,\ell} = \lambda_0 G^\dagger \psi^{\dagger,\ell-1} + S^\dagger.$$

The solution is considered to be converged when the relative, point-wise and group-wise change in the adjoint scalar flux is below a user-defined tolerance:

$$\frac{|\phi_g^{\dagger,(\ell)}(r) - \phi_g^{\dagger,(\ell-1)}(r)|}{\phi_g^{\dagger,(\ell)}(r)} < \epsilon_{\text{tol}}.$$

The rate of convergence turns out to be governed by the dominance ratio, $\mathcal{R} = \frac{\lambda_0}{\lambda_1}$, of the system, much like power iteration. To see this, we write out the first few iterations using an initial guess of zero. The first iteration produces

$$\psi^{\dagger,(1)} = [H^\dagger]^{-1} S^\dagger$$

The second iteration produces

$$\begin{aligned} \psi^{\dagger,(2)} &= [H^\dagger]^{-1} \lambda_0 G^\dagger [H^\dagger]^{-1} S^\dagger + [H^\dagger]^{-1} S^\dagger \\ &= \lambda_0 \sum_{j=0}^{\infty} a_j v_j^\dagger + [H^\dagger]^{-1} S^\dagger \end{aligned}$$

The third iteration produces

$$\begin{aligned}
\psi^{\dagger,(3)} &= [H^\dagger]^{-1} \left[\lambda_0 G^\dagger \psi^{\dagger,(2)} + S^\dagger \right] \\
&= [H^\dagger]^{-1} \left[\lambda_0 G^\dagger \left[\lambda_0 \sum_{j=0}^{\infty} a_j v_j^\dagger + [H^\dagger]^{-1} S^\dagger \right] + S^\dagger \right] \\
&= [H^\dagger]^{-1} \left[\lambda_0 \sum_{j=0}^{\infty} a_j \frac{\lambda_0}{\lambda_j} \lambda_j G^\dagger v_j^\dagger + \lambda_0 G^\dagger [H^\dagger]^{-1} S^\dagger + S^\dagger \right] \\
&= [H^\dagger]^{-1} \left[\lambda_0 \sum_{j=0}^{\infty} a_j \frac{\lambda_0}{\lambda_j} H^\dagger v_j^\dagger + \lambda_0 G^\dagger [H^\dagger]^{-1} S^\dagger + S^\dagger \right] \\
&= \lambda_0 \sum_{j=0}^{\infty} a_j \frac{\lambda_0}{\lambda_j} v_j^\dagger + \lambda_0 \sum_{j=0}^{\infty} a_j v_j^\dagger + [H^\dagger]^{-1} S^\dagger
\end{aligned}$$

Similar steps lead to an expression for the fourth iterate:

$$\psi^{\dagger,(4)} = \lambda_0 \sum_{j=0}^{\infty} a_j \left(\frac{\lambda_0}{\lambda_j} \right)^2 v_j^\dagger + \lambda_0 \sum_{j=0}^{\infty} a_j \frac{\lambda_0}{\lambda_j} v_j^\dagger + \lambda_0 \sum_{j=0}^{\infty} a_j v_j^\dagger + [H^\dagger]^{-1} S^\dagger.$$

The general expression for the N^{th} iterate is

$$\psi^{\dagger,(N)} = [H^\dagger]^{-1} S^\dagger + \lambda_0 \sum_{j=0}^{\infty} a_j v_j^\dagger \sum_{n=0}^{N-2} \left(\frac{\lambda_0}{\lambda_j} \right)^n. \quad (4.8)$$

This expression reveals the importance of eliminating the fundamental mode contamination. To see this, we re-write it as

$$\psi^{\dagger,(N)} = [H^\dagger]^{-1} S^\dagger + \lambda_0 \sum_{n=0}^{N-2} a_0 v_0^\dagger + \lambda_0 \sum_{j=1}^{\infty} a_j v_j^\dagger \sum_{n=0}^{N-2} \left(\frac{\lambda_0}{\lambda_j} \right)^n.$$

If we allow $a_0 \neq 0$, then a non-convergent term appears in the summation. Even if $\|a_0 v_0^\dagger\|$ is arbitrarily small, this term will grow linearly with iteration count and may prevent convergence.

Assuming the fundamental mode contamination is controlled, the exact adjoint flux solution, ψ^\dagger , is reached in the limit of infinite iterations. Using Eq. (4.8), we apply the limit and find

$$\begin{aligned}\psi^\dagger &= \lim_{N \rightarrow \infty} \psi^{\dagger,(N)} = [H^\dagger]^{-1} S^\dagger + \sum_{j=1}^{\infty} \frac{a_j \lambda_0 v_i^\dagger}{1 - \frac{\lambda_0}{\lambda_j}} \\ &= [H^\dagger]^{-1} S^\dagger + \sum_{j=1}^{\infty} b_j v_j^\dagger.\end{aligned}\quad (4.9)$$

Thus, the MSA scheme does converge to the correct solution. From Eq. (4.9), we find that the rate at which the error is reduced from one iteration to the next is limited by $\mathcal{R} = \frac{\lambda_0}{\lambda_1}$, the dominance ratio, or the ratio between the smallest and second-smallest (in magnitude) eigenvalues. Therefore, the computational cost to converge the MSA scheme to a particular tolerance is on the order of the cost to converge the forward power iteration for ψ .

Now we turn to our strategy for controlling the fundamental mode contamination. The initial guess, $\psi^{\dagger,(0)} = 0$ does not contain any fundamental mode. During the iteration process, it is possible that round-off effects or other errors will cause the fundamental mode to creep in. Therefore, after each fixed-source solve during the MSA process (but before the convergence check), we perform an orthogonalization step via the filter

$$\psi^{\dagger,(\ell)} \leftarrow \psi^{\dagger,(\ell)} - \frac{\langle \psi^{\dagger,(\ell)}, G\psi \rangle_{E,\mathcal{D},\Omega}}{\langle v_0^\dagger, G\psi \rangle_{E,\mathcal{D},\Omega}} v_0^\dagger.$$

We note that this filter is equivalent to

$$\psi^{\dagger,(\ell)} \leftarrow \psi^{\dagger,(\ell)} - \frac{\langle \psi^{\dagger,(\ell)}, v_0^\dagger \rangle_{E,\mathcal{D},\Omega}}{\langle v_0^\dagger, v_0^\dagger \rangle_{E,\mathcal{D},\Omega}} v_0^\dagger$$

because of the biorthogonality condition. The former resembles the orthogonality condition that we seek: $\langle \psi^\dagger, G\psi \rangle_{E,\mathcal{D},\Omega} = 0$ and may afford some computational savings because $G\psi$ is isotropic; the latter may have a more intuitive feel for the reader.

Finally, we note that an orthogonalization of S^\dagger can be performed by applying the following filter:

$$S^\dagger \leftarrow S^\dagger - \frac{\langle S^\dagger, \psi \rangle_{E,\mathcal{D},\Omega}}{\langle \psi, G^\dagger v_0^\dagger \rangle_{E,\mathcal{D},\Omega}} G^\dagger v_0^\dagger.$$

In theory, this calculation is not necessary because S^\dagger is analytically orthogonal to ψ due to the form of our QOIs. In practice, however, it may be necessary to eliminate fundamental mode contamination from round off effects or other small errors in the system unknowns. The orthogonalization may be completed prior to the MSA solve and does not need to be repeated at each iteration.

As a review, Fig. 4.4 gives the pseudocode for the MSA solver. The initial guess for ψ^\dagger is either zero for the first MSA solve or the previous, converged ψ^\dagger . The homogeneous adjoint solution is computed before entering the MSA solve using regular power iteration. We compute, orthogonalize, and store S^\dagger , whose form depends on the particular QOI. We then enter the MSA iterations, which either terminate after a maximum number of iterations or after the convergence criteria is met.

- given: initial guess $\psi^{+,0}$; homogeneous adjoint solution v_0^+ ;
- compute and orthogonalize S^+ ;
- $i=0$;
- while ($\text{!converged \&\& } i < \text{maxIter}$)
 1. compute source: $S^{+,i} = S^+ + \lambda G^+ \psi^{+,i}$;
 2. compute $\psi^{+,i+1}$: $H^+ \psi^{+,i+1} = S^{+,i}$;
 3. orthogonalize $\psi^{+,i+1}$;
 4. converged = check-convergence;
 5. $i++$;

Figure 4.4: Pseudocode for the MSA solver

4.5 Description of the Depletion-Perturbation Solver in PDT

In this section, we give an overview of the primary computational workhorse of this dissertation: the depletion perturbation solver, or DplAdjointSolver. This solver begins by computing a forward depletion solution (that is, $N(t)$ and $\psi(t)$). Using schemes that we describe below, this solution is either stored or checkpointed so that it can be made available during the subsequent adjoint solve. Next an adjoint time integration is launched using an appropriate terminal condition, and the proper form of the sensitivity and/or error equation is integrated to compute the desired UQ information.

The solver can be described in terms of two major efforts. First is a time marching scheme to integrate the adjoint variables backwards in time from their terminal condition at $t = t_f$ to the starting point of the simulation, $t = t_0$. The terminal condition is inherently tied to the particular QOI and may require an intermediate adjoint transport solve (see Eq. (3.84) for an example). From this condition, the adjoint equations are traced backwards over the same time steps taken by the forward solver. As described in Sec. 2.5.2, the rule for integrating the adjoint equations over the time step $t \in [t_n, t_{n-1}]$ is the adjoint rule of the forward Runge-Kutta scheme.

We also integrate the sensitivity equation (e.g. Eq. (3.85)) during the adjoint solve. This is computationally advantageous because evaluation of the cross-correlation terms at a given stage requires access to the full adjoint and forward solution, both of which are available during the adjoint solve. The alternative would be to checkpoint and recompute these solutions at quadrature points after the adjoint solve is complete. Our scheme, as described in Sec. 3.5, integrates the sensitivity equation with the same Runge-Kutta scheme (and therefore same order of accuracy) as the adjoint equations.

Pseudocode for the procedure of integrating the adjoint equations and sensitivity expression over a single time step is given in Fig. 4.5.

- given access to forward solution: $\psi(t_i), N(t_i)$, $i=1\dots nStage$;
- given ψ^+_n, N^+_n
- for $s=1:nStage$
 1. if ($s > 1$)
 - compute ψ^+_s ;
 2. compute and store $(dN^+/dt)_s$;
 3. evaluate stage s cross-correlation terms
 4. if ($s < nStage$)
 - compute N^+_{s+1} ;
- compute $N^+(t_{n-1})$;
- compute $\Phi^+(t_{n-1})$;
- integrate sensitivity equation from t_{n-1} to t_n

Figure 4.5: Pseudocode for integrating the adjoint equations

For example, assume we are using a self-adjoint Runge-Kutta scheme such that

$b^\dagger = b$ and $a^\dagger = a$. Further, assume that our sensitivity equation has the form

$$\frac{dQ}{dp} = \int_{t_0}^{t_f} \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi] \right\rangle_{E,\mathcal{D},\Omega} dt.$$

Then the adjoint k-eigenvalue depletion equations (those corresponding to forward problem Eqs. (3.54) – (3.57)) and the sensitivity equation are advanced *backwards* in time from t_n to t_{n-1} according to the rule

$$\begin{aligned} N_{n-1}^\dagger &= N_n^\dagger - \Delta t \sum_{i=1}^s b_i \left(\frac{dN^\dagger}{dt} \right)_i \\ \frac{dQ}{dp} &+ = \Delta t \sum_{i=1}^s b_i \mathcal{C}_i \\ H_i^\dagger \psi_i^\dagger - \lambda_i G_i^\dagger \psi_i^\dagger &= S_i^\dagger \\ \left\langle \psi_i^\dagger, G_i \psi_i \right\rangle_{E,\mathcal{D},\Omega} &= 0 \\ \mathcal{C}_i &= \left\langle \psi_i^\dagger, \frac{\partial}{\partial p} [H_i \psi_i] \right\rangle_{E,\mathcal{D},\Omega} \\ N_i^\dagger &= N_n^\dagger - \Delta t \sum_{j=1}^{i-1} a_{ij} \left(\frac{dN^\dagger}{dt} \right)_j. \end{aligned}$$

Here, operators operator subscript i indicates that the operator may contain terms from both the forward and adjoint solutions evaluated at stage i . Again, this is a key computational challenge associated with the adjoint technique: how can we manage the time series of a large forward solution such that pieces of it are available as required during the adjoint solve?

This question is related to the second major effort of this solver: providing access to, or checkpointing, the forward solution (N and ψ) in order to compute the operators and source terms that appear in the adjoint and sensitivity equations. In the following section, we outline and analyze in detail a family of checkpointing al-

gorithms. Here, we introduce a basic checkpointing algorithm and describe some of the algorithmic trade-offs that must be considered in an application.

Our overarching goal in designing a checkpointing algorithm is to minimize time to solution. We divide the time to solution into computational time, or time spent computing the forward and adjoint depletion solution, and time spent writing and reading checkpointing files to and from disk.

For the depletion perturbation problem, we find that greater than 95% of the clock time is spent performing fixed-source transport iterations (computing the flux solution). Therefore, we use the number of transport solves (either forward or adjoint, fixed-source or k-eigenvalue) as the primary computational cost of any scheme. The variation of this cost from scheme-to-scheme will be in the number of times that the forward solution must be re-computed. This variation is due to the available RAM footprint for storing these re-computes as well as to exactly what is stored to represent the forward solution. A secondary cost will be the number of “recovery sweeps”, which we will define in the subsequent section.

For disk cost, we tabulate the total number of bytes (or floating point unknowns) that must be written to and read from files on disk. Again, the variability in this cost will depend on the number of time steps between checkpoints and the amount of information that is written to disk in order to represent the full solution.

An illustration of a general checkpointing algorithm is presented in Figure 4.6. The time domain is broken into re-compute segments. For our depletion problems, the segments consist of some number of depletion cycles, and the number of cycles in a segment depends on the amount of memory available for storing the forward solution within a segment. First, the forward problem is solved to completion. During this solve, a snapshot of the depletion solution is written to disk at the beginning and end of each re-compute segment. As the figure indicates and we describe later in the text,

some savings can be afforded by treating the last re-compute segment differently.

During the adjoint solve, the code switches between re-compute mode and adjoint mode. For each re-compute segment (except for the last), the solver must first use the printed-to-file snapshots at the beginning of the segment to re-compute and store the forward solution at each time-node within that segment. Then the solver enters adjoint mode and uses the available forward solution information to form and solve the adjoint equations.

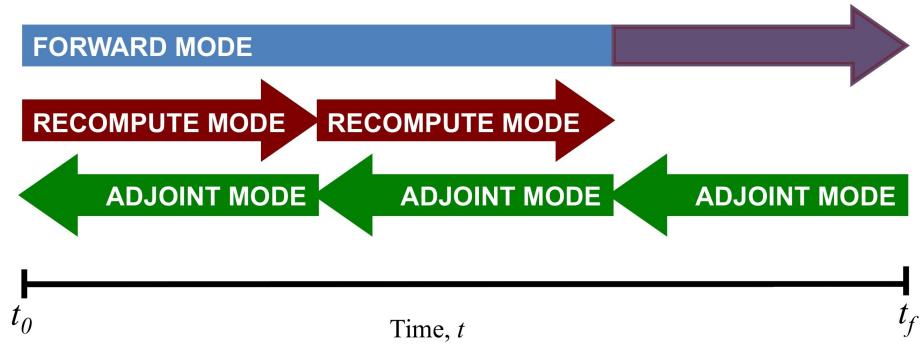


Figure 4.6: Illustration of a general checkpointing scheme

We already see some of the trade-offs that must be considered in the design of a checkpointing algorithm. The re-compute ratio is the percentage of the forward solution that must be re-computed during the adjoint solve. For a scheme with a relatively large number of short re-compute segments, the re-compute ratio is smaller because more of the forward solution can be read from file; such a scheme also has a smaller memory footprint, but incurs a larger file I/O cost. A scheme with fewer, longer re-compute chunks will have a higher re-compute ratio and RAM footprint, but will spend less time in I/O mode. The characteristics of the machine will determine an optimal re-compute ratio for a given problem.

4.6 Schemes for Checkpointing the Forward Solution

In this section, we motivate and describe a family of checkpointing algorithms and their implementation in the PDT code. The checkpointing algorithms are designed to efficiently provide access to the correct snapshot of the forward solution during the integration of the adjoint and sensitivity equations. The novelty in the new schemes is reduced I/O load on the machine by taking advantage of the spherical harmonics representation of the transport source.

We first motivate the need for efficient checkpointing schemes by describing the memory footprint of an example high-fidelity reactor-analysis problem on a state-of-the-art architecture. In the second subsection, we present a new option to consider for checkpointing: in addition to choosing the checkpointing frequency, we describe a choice of *what* to checkpoint in order to represent the forward solution. For the transport solution, we show that a lower dimensional representation is available. In the remaining subsections, we describe our checkpointing algorithms and analyze their cost in terms of transport solves, sweep counts, memory footprint, and I/O load.

4.6.1 Motivation for efficient checkpointing: An example problem

It is not difficult to imagine a problem where the full time-series of the forward solution will not fit on the RAM of the host machine. Consider a high-fidelity reactor simulation on the Sequoia Blue Gene system at Lawrence Livermore National Laboratory. This machine has about 100,000 nodes, each with 16 CPUs and 16GB of shared RAM, an example of the future architectures with increasing core counts and decreasing RAM capacity per cpu.

First, suppose that the geometry, cross sections, and the remainder of the problem definition take half of the available RAM per node. Further, suppose we domain

decompose such that each node has 1000 spatial cells, each with 4 spatial elements (using linear finite elements). If we are using 200 energy groups and 500 angles (which are moderately high fidelity by today’s standards), then a snapshot of the angular flux using double-precision costs 3.2GB of RAM per node. At this point we could only afford to store a few such flux snapshots, and we haven’t even considered the memory footprint of the adjoint problem and forward/adjoint densities!

Clearly, simply storing the entire time-series of the forward solution is not an option for realistic problems. Instead, it is common practice to checkpoint, or write snap-shots of the forward solution to file during the forward solution. Then, as necessary during the adjoint solution, these checkpoints are read back into RAM, and the required “chunk” of the forward solution is re-computed and used to solve the corresponding chunk of the adjoint solution; then it is thrown away. Even yet, repetitively reading and writing chunks of data 3.2GB at a time on thousands of nodes is a costly I/O task and will limit the efficiency of high-fidelity adjoint calculations.

4.6.2 Checkpointing source moments as a representation of the angular flux

One contribution of this dissertation is an idea that decreases the memory footprint and I/O load of a single checkpoint at the cost of extra floating point operations (FLOPs). This is exactly the sort of tradeoff we need for efficient use of the advanced architectures that appear to be in our future. Our scheme takes advantage of the spherical-harmonic representation of the angular dependence of the scattering source. The inscatter source in the transport equation is

$$S_S(r, E, \Omega) = \int dE' \int d\Omega' \Sigma_s(r, E' \rightarrow E, \Omega' \rightarrow \Omega) \psi(r, E', \Omega').$$

It is common practice to represent this source in terms of a truncated expansion in spherical harmonic functions, Y , with coefficients C , as follows

$$S_S(r, E, \Omega) = \int dE' \sum_{k=0}^{\mathcal{K}} C_k \Sigma_s(E' \rightarrow E) Y_k(\Omega) \int d\Omega' Y_k(\Omega') \psi(r, E', \Omega').$$

The advantage of this representation is that the number of moments required to sufficiently capture the source anisotropy is typically much less than the number of discrete ordinates being modeled. Thus, only $\mathcal{K} + 1$ scalar flux moments,

$$\phi_k = \int d\Omega' Y_k \psi(r, E', \Omega'),$$

must be stored per spatial element, per group, per sweep, as opposed to the full angular flux vector, in order to form the scattering source.

Along the same vein, we realize that the angular flux vector can be re-created with a single sweep if all of the total source moments are available. Thus, if a checkpoint only writes to disk the total source moments, then the cost we pay to recreate the angular flux at a later time is that to read back in the total source moments and perform one sweep. In other words, we incur modest extra computational cost in order to reduce the data throughput to and from disk.

This tradeoff is aligned with the hardware tradeoffs that are enabling machines to perform at increasingly higher FLOP rates. The trend is that FLOPs are becoming cheaper, but memory availability (per processor) and file I/O bandwidth (per processor) are decreasing[4, 5]. This does not bode well for classic checkpointing algorithms, as their performance improves with increased memory availability and I/O bandwidth. By taking advantage of this lower dimensional representation of the source, we are more likely to scale well on advanced architectures.

4.6.3 Computational analysis of five checkpointing schemes

In this subsection, we present an analysis of five checkpointing schemes. Three of these schemes will read, write, and/or store only the converged source moments to represent the forward flux solution. Of course, these schemes will incur a “recovery sweep” cost, or the cost of a sweep that is used to re-create a flux solution from the stored source moments. In addition to this cost, we also count the total number of transport solves and the data throughput to and from disk.

The analysis is presented for the source-driven case because each transport solve is a single fixed-source solve, which is easier to follow. We will comment on the k-eigenvalue case near the end of the section. We consider a multi-cycle depletion problem. Each cycle consists of several time steps, and each time-step may contain multiple stages. We use the following notation to model the cost of each scheme:

- N : The total number of depletion cycles to be run in the forward problem;
- K : The number of depletion cycles per re-compute segment;
- T : The number of time-steps per cycle;
- S : The number of stages per time-step;
- M_ψ : The memory footprint of a single copy of the angular flux vector;
- M_S : The memory footprint of a single copy of the converged source moments.

For simplicity in this analysis, we will assume that the ratio $N_R = N/K$, the number of recompute segments, is an integer. The number of stages per time-step is important because a fixed-source solve is required at each stage. We focus on the checkpointing and re-computation of the neutron flux because it is the more expen-

sive in terms of memory load and computation time; the reader must note, however, that storage and re-computation of the Bateman solution is also required.

To represent the schemes, we use illustrations similar to Fig. 4.6 except that we use symbols to denote the work required at each stage. Figure 4.7 will serve as a legend for these illustrations.

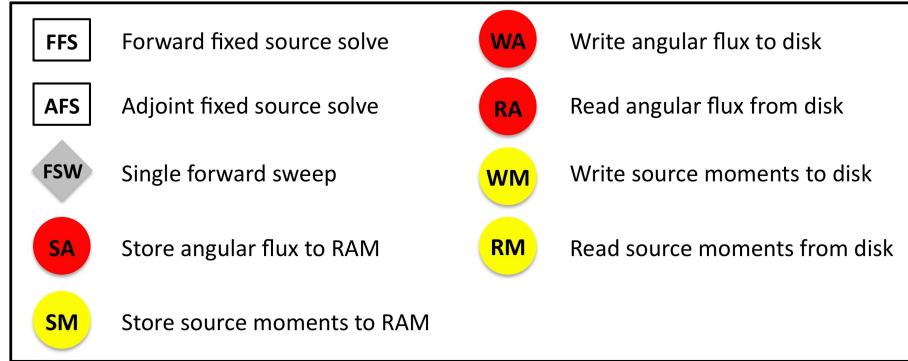
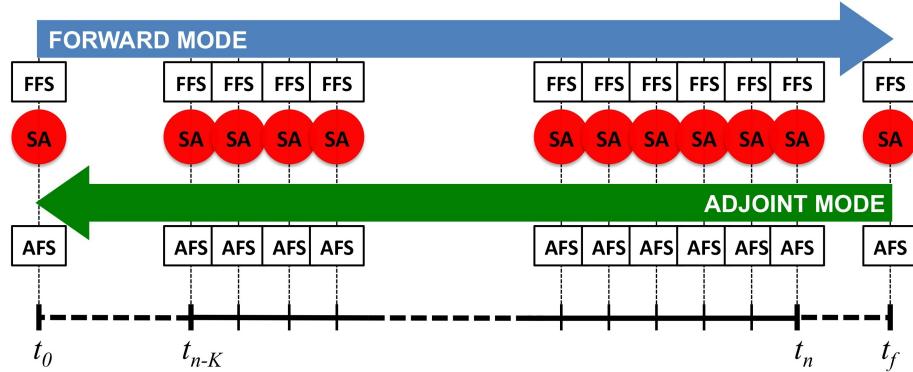
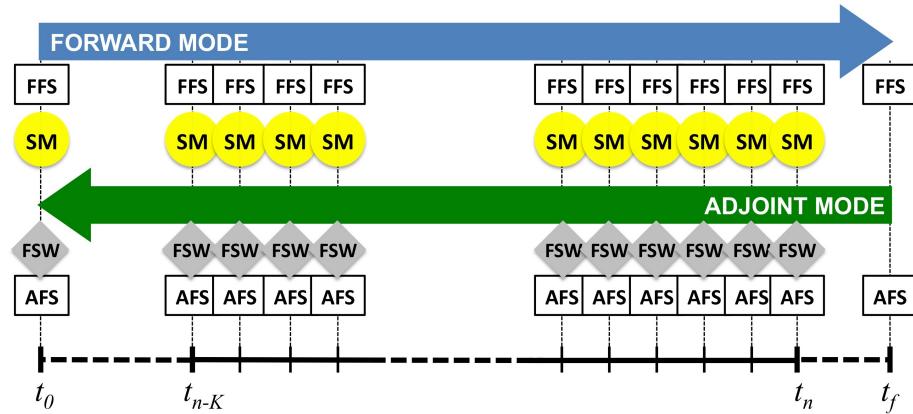


Figure 4.7: Legend of symbols for checkpointing scheme cost analysis

To help introduce our notation, we first describe two schemes that store the entire forward solution during the forward solve, thereby eliminating the need for file I/O and re-compute mode during the adjoint solve. These schemes are illustrated in Fig. 4.8.



(a) STOR_ALL Scheme



(b) STOR_MOM Scheme

Figure 4.8: Illustration of the STOR_ALL and STOR_MOM schemes.

Figure 4.9(a) represents a scheme where the entire angular flux vector is stored at each stage of each time step of each cycle during the forward solve. During the adjoint solve, no work is required to re-compute the forward solution. The number of forward and adjoint fixed-source solves is $S \cdot T \cdot N + 1$ each. There are no re-compute fixed source solves. The RAM footprint is $M_\psi + M_\psi(S \cdot T \cdot N + 1)$. The first M_ψ is allocated to store ψ^\dagger after it is calculated at each time step; the remaining $S \cdot T \cdot N + 1 - M_\psi$ allocations are for storing all copies of ψ . We call this scheme STOR_ALL.

To implement this scheme, we simply pre-compute the number of snapshots of

ψ that we will be required to store, and we allocate the necessary space for this storage on each spatial element. Then, after each forward fixed-source solve, we simply save the current ψ in the allocated space, where it stays until the end of the entire calculation.

Figure 4.9(b) illustrates a scheme where only the converged source moments are stored at each stage of each time step of each cycle during the forward solve. As a result, a single forward sweep is required at each stage during adjoint mode before the adjoint fixed source solve. Again, the cost of the forward and adjoint solves is $S \cdot T \cdot N + 1$ fixed-source solves each. The re-compute cost is $S \cdot T \cdot N$ sweeps, as no sweep is required to re-compute $\psi(t_f)$. We call this scheme STOR_MOM.

The RAM footprint of the STOR_MOM scheme is $2M_\psi + M_S(S \cdot T \cdot N + 1)$. The two M_ψ allocations come from the requirement to keep a single copy of ψ (after a recovery sweep) and ψ^\dagger at each time step. The $S \cdot T \cdot N$ allocations of M_S vectors are required to store the forward source moments. The last M_S vector allocation is used to store the adjoint source moments during a forward recovery sweep. This allows the adjoint calculation to pick up where it left off after the previous fixed source solve, hopefully reducing the number of adjoint source iterations.

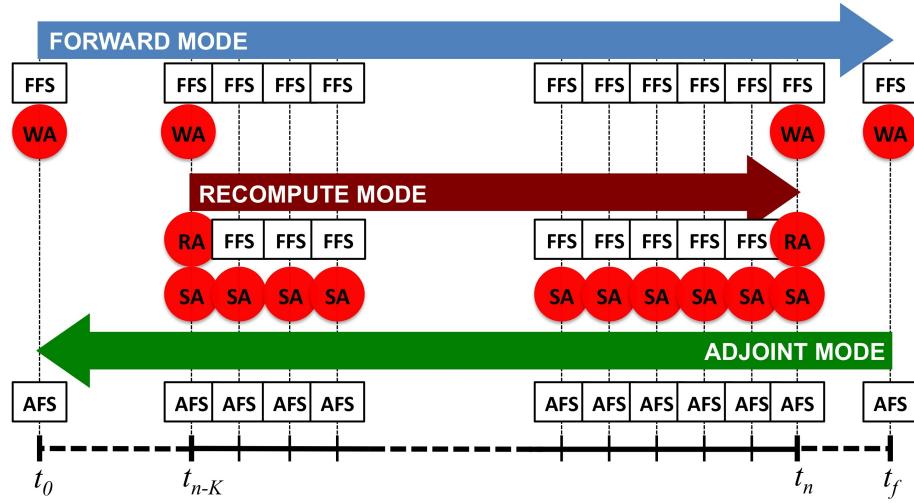
The implementation of STOR_MOM is similar to STOR_ALL in that we pre-compute the number of source moments snapshots that each element must store and allocate that space. The converged source moments are simply copied into the allocated space after each forward fixed source solve. During the adjoint solve, when the forward flux is needed, the adjoint source moments are moved out of the active source vector, the proper forward source moments are moved into the active source vector, a sweep is performed, ψ is saved, and the adjoint source moments are moved back in as the initial guess for the next adjoint fixed-source solve.

Conceptually these checkpointing schemes are very straightforward. The STOR_ALL

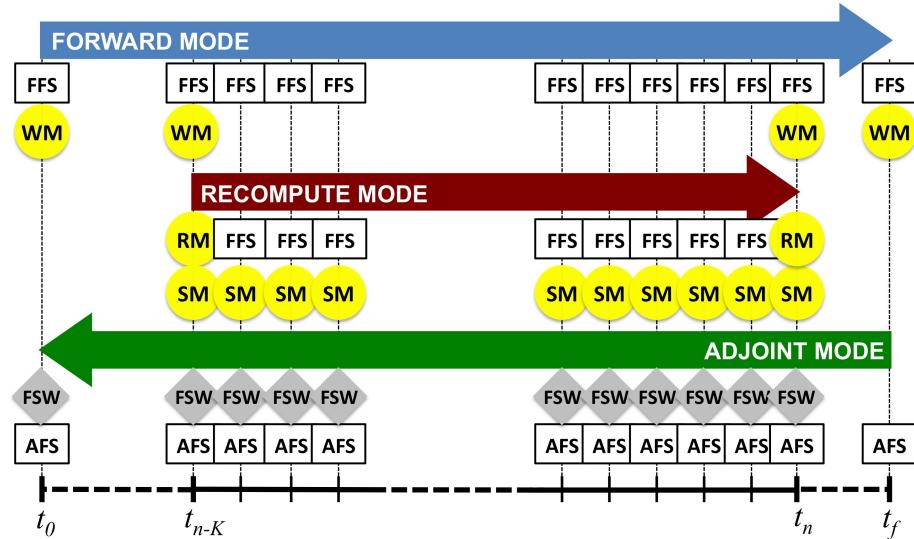
scheme, however, is simply not an option for large reactor simulations. Alternatively, the STOR_MOM scheme may be an option for some medium-size simulations. Its RAM footprint depends on \mathcal{K} , the degree of anisotropy being modeled, and the number of time-steps in the simulation.

The depletion perturbation problems that we are targeting will certainly require some interaction with the disk to manage access to the forward solution. First we present two schemes that are analogs to STOR_ALL and STOR_MOM. They are CKPT_ALL and CKPT_MOM, illustrated in Fig. 4.9.

On the time axes in Fig. 4.9, we highlight the re-compute segment consisting of $S \cdot T \cdot K + 1$ stages between t_{n-K} and t_n . Note that this is not the last re-compute segment. We will address our treatment of this segment below. During the forward solve of the CKPT_ALL scheme, the full angular flux vector is written to binary file at the beginning/end of each recompute segment. Then, during adjoint mode, when the solver reaches t_n , $\psi(t_{n-K})$ and $\psi(t_n)$ are read from file and stored to RAM. The solver then enters recompute mode and performs $S \cdot T \cdot K - 1$ forward fixed-source solves, saving ψ to RAM after each solve. Once the forward solution is recomputed to t_n , the solver enters adjoint mode and solves for ψ^\dagger and N^\dagger backwards to t_{n-K} . Because the full angular flux vector is sitting in RAM, no further recompute cost is incurred during the adjoint solve.



(a) CKPT_ALL Scheme



(b) CKPT_MOM Scheme

Figure 4.9: Illustration of the CKPT_ALL and CKPT_MOM schemes.

The CKPT_MOM scheme has two major differences from the CKPT_ALL scheme. First, while in forward mode, the converged source moments are written to file at the beginning and end of each recompute segment. Compared to writing ψ , this may be a substantial decrease in the I/O load. Second, during the recompute mode,

the source moments are saved to RAM instead of the full angular flux; this is a substantial decrease in RAM footprint. The price we pay for the decreased I/O load and RAM footprint is an extra forward sweep at each stage during the adjoint solve.

We can predict the number of fixed source solves and single sweeps required to complete the CKPT_ALL and CKPT_MOM schemes. First, there are again $S \cdot T \cdot N + 1$ of the forward and adjoint fixed source solves, each. This cost should not change as the recompute strategy is changed. The number of recompute fixed source solves depends on N_R , which is the total number of recompute segments. The forward and recompute modes are combined during the terminal K cycles of the forward problem; in other words, these cycles do not have to be recomputed because the necessary information is saved during the forward mode. The beginning and end of segment fluxes are read from file for each of the remaining $N_R - 1$ segments. We find that the total number of recompute fixed source solves for the CKPT_ALL and CKPT_MOM schemes is $(N_R - 1)(S \cdot T \cdot K - 1)$.

There are some subtleties not illustrated in the figure that must be considered when predicting the number of single sweeps. The CKPT_ALL scheme does not require any single sweeps because the full ψ is always stored. For the CKPT_MOM scheme, all stages require at least one sweep to reconstruct ψ during adjoint mode except for $\psi(t_f)$. This is $S \cdot T \cdot N$ single sweeps. Single sweeps are also required during each of the $N_R - 1$ recompute segments. One kicks off the recompute; another comes at the end, where ψ is recovered with a single sweep instead of using a fixed-source solve. We find that the scheme requires a total of $S \cdot T \cdot N + 2(N_R - 1)$ single forward sweeps. As in the CKPT_MOM scheme, we also require N_R single adjoint sweeps to restore the adjoint state after a recompute segment.

The RAM footprint of these schemes is fairly straightforward to compute. The number of forward snapshots saved during a recompute segment is $S \cdot T \cdot K + 1$. We

also allocate space for a transition vector to store the adjoint state during recompute mode, and vice-versa. The RAM footprint for CKPT_ALL is therefore $M_\psi + M_\psi(S \cdot T \cdot K + 2)$. Again, for the CKPT_MOM vector, we must allocate space for one full ψ snapshot to store the forward flux after the single recomputes. Therefore, its RAM footprint is $2M_\psi + M_S(S \cdot T \cdot K + 2)$.

Lastly, we introduce a scheme whose RAM footprint (for the flux solution) does not grow with S , T , or K . The scheme mimics the CKPT_MOM scheme, but does not save the source moments to RAM during the recompute mode. Instead, when ψ is needed during the adjoint solve, we re-solve a fixed source problem using an interpolated guess for the initial source moments; we call this scheme INTP_MOM, and it is illustrated in Fig. 4.10.

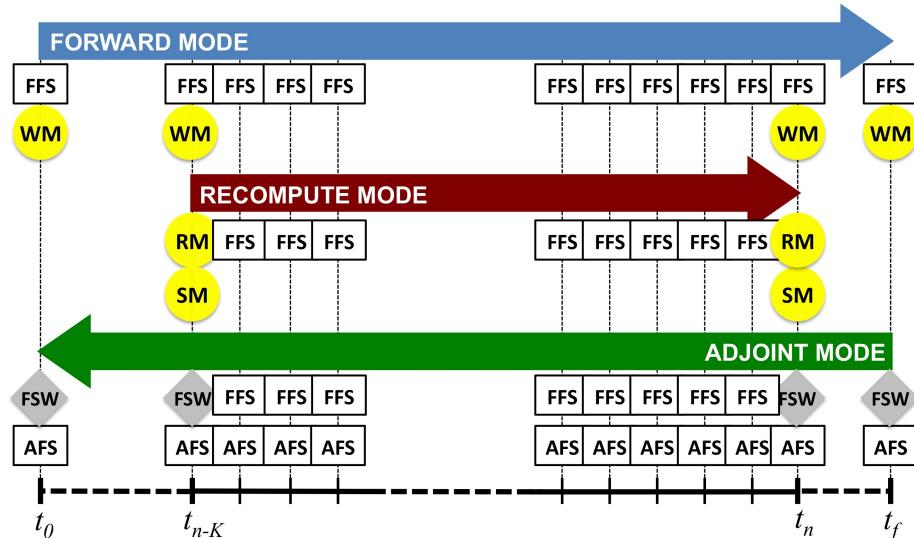


Figure 4.10: Illustration of the INTP_MOM checkpointing scheme

Like the others, this scheme requires the minimum $S \cdot T \cdot N + 1$ initial forward and adjoint fixed source solves, each. The recompute mode is the same as that for the CKPT_MOM scheme except that the converged source moments are not

stored to RAM; instead, only the forward densities are stored to RAM during the recompute. Then, during the adjoint solve, a single forward sweep is required to reconstruct $\psi(t_{n-K})$. The remaining ψ vectors inside the recompute segment are solved using another full fixed source solve. The initial guess for the source moments is interpolated linearly from t_{n-K} and t_n . This applies to the last recompute segment as well, which must be re-created during the adjoint solve. Therefore, the number of fixed source solves is $2(N_R - 1)(S \cdot T \cdot K - 1) + (S \cdot T \cdot K - 1) = (2N_R - 1)(S \cdot T \cdot K - 1)$.

The number of single forward sweeps is $3(N_R - 1) + 1$. Two sweeps are required per recompute segment; that totals $2(N_R - 1)$. One sweep is required per adjoint mode to construct $\psi(t_{n-K})$. So the total number of forward sweeps is $2(N_R - 1) + N_R = 3(N_R - 1) + 1$. As in the CKPT_MOM scheme, N_R single adjoint sweeps are required to restore the adjoint state after a forward recompute segment.

The total computational cost of this scheme is close to twice the cost of CKPT_MOM, but it has a substantially smaller RAM footprint. The allocation for flux storage is $2M_\psi + 3M_S$, independent of the number of time-steps per cycle and the number of cycles per checkpoint. Of course, the allocation for storing the forward densities does grow with these factors, but the number of density unknowns is typically much, much smaller than the number of flux or moment unknowns. This means that the RAM allocation for checkpointing may be able to handle many, many more cycles, potentially reducing the I/O load to a very small number of disk accesses. For advanced architectures where not all nodes have direct access to the file system, this could avoid substantial network bandwidth costs.

The final cost consideration for these schemes is the data throughput to and from disk. The CKPT_ALL scheme reads/writes M_ψ data per file, while the CKPT_MOM and INTP_MOM schemes read/write M_S data per file. Although its not explicitly necessary to write a file at $t = t_f$, we do write the file because we sometimes run

in “adjoint-only mode,” where the forward solution is already in files and therefore is not computed. Thus, the total data throughput to disk is $(N_R + 1)M_\psi$ for the CKPT_ALL scheme and $(N_R + 1)M_S$ for the CKPT_MOM and INTP_MOM schemes. The same throughput cost is incurred from disk.

Note that these throughputs are *per core*; that is, each core writes its own snapshot files. This brings forth many questions about the proper way to handle parallel I/O on massively parallel machines. Much of this discussion will fall outside the scope of this work; however, we have implemented options for creating multiple directories and file names for this I/O load in PDT. These options, along with all other details of the depletion, adjoint, and checkpointing implementations, are documented in Appendix B.

In summary, we have developed and analyzed the computational and I/O cost of a family of checkpointing algorithms, and we have implemented the schemes in PDT. Three of the schemes leverage a novel idea for reducing I/O load by only checkpointing the converged source moments in the transport equation, as opposed to checkpointing the entire angular flux vector. These schemes do incur extra FLOP costs; we hypothesize, however, that this tradeoff will reduce overall time to solution for large, parallel problems. We will perform computational experiments to verify this hypothesis in later sections of this dissertation.

Table 4.1 summarizes the fixed source cost of each scheme, and Table 4.2 summarizes the single sweep cost, RAM footprint, and total data-to-disk throughput of each scheme.

Table 4.1: Number of forward, adjoint, and recompute fixed source solves for each checkpointing scheme

Scheme	Forward Fixed Source Solves	Adjoint Fixed Source Solves	Recompute Fixed Source Solves
STOR_ALL	$S \cdot T \cdot N + 1$	$S \cdot T \cdot N + 1$	0
STOR_MOM	$S \cdot T \cdot N + 1$	$S \cdot T \cdot N + 1$	0
CKPT_ALL	$S \cdot T \cdot N + 1$	$S \cdot T \cdot N + 1$	$(N_R - 1)(S \cdot T \cdot K - 1)$
CKPT_MOM	$S \cdot T \cdot N + 1$	$S \cdot T \cdot N + 1$	$(N_R - 1)(S \cdot T \cdot K - 1)$
INTP_MOM	$S \cdot T \cdot N + 1$	$S \cdot T \cdot N + 1$	$(2N_R - 1)(S \cdot T \cdot K - 1)$

Table 4.2: Single sweep cost and RAM footprint of the checkpointing schemes

Scheme	Forward Single Sweeps	Adjoint Single Sweeps	RAM Footprint
STOR_ALL	0	0	$M_\psi + M_\psi(S \cdot T \cdot N + 1)$
STOR_MOM	$S \cdot T \cdot N$	0	$2M_\psi + M_S(S \cdot T \cdot N)$
CKPT_ALL	0	0	$M_\psi + M_\psi(S \cdot T \cdot K + 2)$
CKPT_MOM	$S \cdot T \cdot N + 2(N_R - 1)$	N_R	$2M_\psi + M_S(S \cdot T \cdot K + 2)$
INTP_MOM	$3(N_R - 1) + 1$	N_R	$2M_\psi + 3M_S$

The preceding cost analysis of the checkpointing schemes was performed for the case of the source-driven depletion equations. The case of the k-eigenvalue formulation is more costly in terms of computation and memory load. First, instead of $S \cdot T \cdot N + 1$ fixed source solves for the forward and adjoint modes, we require

$S \cdot T \cdot N + 1$ power iteration solves and MSA solves, respectively, for the forward and adjoint modes. The homogeneous adjoint k-eigenvalue equation must also be solved before each MSASolve – or $S \cdot T \cdot N + 1$ times – in order to orthogonalize ψ^\dagger to the forward fission source. This orthogonalization also requires that each scheme allocate one extra M_ψ vector to store the adjoint fundamental mode flux. The number of forward and adjoint recovery sweeps remains unchanged, as does the data throughput.

5. VERIFICATION AND TEST PROBLEMS

In this section, we design, implement, and discuss a suite of verification problems. Per the findings of a recent National Academy of Sciences report[3], we design our verification suite around the hierarchical nature of the depletion perturbation problem. In other words, we decompose the code into testable components: the depletion solver, the adjoint transport solver, and finally the depletion perturbation solver.

Our tests take one of two forms. First, when possible, we compare numerical PDT solutions to analytic solutions. Analytic transport solutions are only available in simple limits, such as in an infinite medium, which we use in several of the problems that follow. Analytic solutions to more interesting and realistic problems, however, are difficult or impossible to develop. Therefore, the second set of verification problems rely on the demonstration of expected rates of convergence or on comparison with solutions to other codes.

In the first section, we test the depletion solver with two problems. First, we compare the PDT solution to an analytic solution in an infinite medium, two group problem. In the second problem, we simulate the Plutonium-239 production chain in a 33 group, infinite medium problem, and compare our solution to a solution generated by the Matlab ODE solver suite. In the second section, we document two problems designed to test the steady-state adjoint transport solver. The first is an infinite medium problem with analytic solution, and the second is a detector response problem where we test the convergence rates of the computed derivatives.

In the last two sections, we test the depletion perturbation solver. In the first, we perform a detailed walk-through of an infinite medium, k-eigenvalue depletion perturbation problem, comparing expected and computed numerical results. We then

compare these computed results with analytic QOI sensitivity and error calculations, and show that the schemes are producing the expected rates of convergence. In the final section, we verify that the computed parameter derivatives predict QOI perturbations at the expected order of accuracy for a source-driven problem.

As a result of verification exercise described in this section, we have an increased level of confidence that the implementation of the Bateman solver, adjoint transport solver, and depletion perturbation solver in PDT are solving their respective equations correctly. Moreover, the hierarchical approach gives insight into the individual behaviors of the components, and allowed us to more efficiently chase, find, and squash bugs in the code.

5.1 Verification Problems Targeting the Depletion Solver

In this section, we describe verification problems designed to test the depletion solver in PDT. We first test the depletion solver in the limit of an infinite medium. We then compare a PDT depletion solution to a reference solution generated in Matlab for a simulation of the Plutonium-239 production chain.

5.1.1 A single component, infinite medium depletion verification problem in k-eigenvalue form

We develop an analytic solution to a k-eigenvalue depletion problem with a single component and two energy groups. The component has cross sections

$$\begin{aligned}\sigma_{t,1} &= \sigma_{R,1} + \sigma_{s,1 \rightarrow 1} + \sigma_{s,1 \rightarrow 2} \\ \sigma_{t,2} &= \sigma_{\gamma,2} + \sigma_{f,2} + \sigma_{s,2 \rightarrow 1} + \sigma_{s,2 \rightarrow 2} \\ &= \sigma_{R,2} + \sigma_{s,2 \rightarrow 1} + \sigma_{s,2 \rightarrow 2}\end{aligned}$$

decay constant λ , energy per fission E_f , neutrons per fission ν , and fission spectrum χ_1 and χ_2 . The variable σ_R is the sum of all interactions that result in the destruction of the nuclide. The infinite medium operates at a constant power density of \mathcal{P}_0 . The governing depletion equations in an infinite medium are

$$\begin{aligned}\frac{dN}{dt} &= -4\pi ANF_b^{\text{cm}} [\sigma_{R,1}\psi_1 + \sigma_{R,2}\psi_2] - \lambda N \\ \sigma_{t,1}N\psi_1 - N(\psi_1\sigma_{s,1 \rightarrow 1} + \psi_2\sigma_{s,2 \rightarrow 1}) - \frac{1}{k_{\text{eff}}}\chi_1\nu N\sigma_{f,2}\psi_2 &= 0 \\ \sigma_{t,2}N\psi_2 - N(\psi_1\sigma_{s,1 \rightarrow 2} + \psi_2\sigma_{s,2 \rightarrow 2}) - \frac{1}{k_{\text{eff}}}\chi_2\nu N\sigma_{f,2}\psi_2 &= 0 \\ \mathcal{P}_0 - A4\pi N E_f \sigma_{f,2} \psi_2 &= 0 \\ N(0) = N_0, \quad t \in [0, T].\end{aligned}$$

After some manipulation of the transport equations, we find the following fundamental mode (more detail is provided in Sec. 5.3):

$$\begin{aligned}\frac{\psi_1}{\psi_2} &= \frac{\frac{1}{k_{\text{eff}}}\chi_1\nu\sigma_{f,2} + \sigma_{s,2\rightarrow 1}}{\sigma_{t,1} - \sigma_{s,1\rightarrow 1}} \\ \frac{1}{k_{\text{eff}}} &= \frac{(\sigma_{t,1} - \sigma_{s,1\rightarrow 1})(\sigma_{t,2} - \sigma_{s,2\rightarrow 2}) - \sigma_{s,1\rightarrow 2}\sigma_{s,2\rightarrow 1}}{\nu\sigma_{f,2}[\chi_1\sigma_{s,1\rightarrow 2} + \chi_2(\sigma_{t,1} - \sigma_{s,1\rightarrow 1})]}.\end{aligned}$$

Substituting this expression and the power constraint into the Bateman equation, we find

$$\begin{aligned}\frac{dN}{dt} &= -\frac{\mathcal{P}_0 F_b^{\text{cm}}}{E_f \sigma_{f,2}} \left[\sigma_{R,1} \frac{\frac{1}{k_{\text{eff}}}\chi_1\nu\sigma_{f,2} + \sigma_{s,2\rightarrow 1}}{\sigma_{t,1} - \sigma_{s,1\rightarrow 1}} + \sigma_{R,2} \right] - \lambda N \\ &\equiv -\kappa - \lambda N.\end{aligned}\tag{5.1}$$

Equation (5.1) has analytic solution

$$N(t) = -\frac{\kappa}{\lambda} + \left(N_0 + \frac{\kappa}{\lambda} \right) \exp(-\lambda t).\tag{5.2}$$

Finally, suppose that our QOI is the mol density ($\left[\frac{\text{mol}}{\text{cm}^3} \right]$) of the component at $t = T$, computed as

$$Q = \frac{N(T)}{N_A}$$

where N_A is Avogadro's number in units of $\left[\frac{\text{atom-cm}^2}{\text{b-mol}} \right]$. We devised a series of numerical test problems to verify that the solution given by the depletion solver approaches this analytic QOI at the rate expected from the particular Runge-Kutta scheme in use. We tested the explicit Euler, implicit Euler, modified Euler, and Runge-Kutta 4 schemes (see Appendix E). The problem parameters are defined in Table 5.1.

Table 5.1: Parameters for infinite medium, 2 group k-eigenvalue depletion verification problem

Parameter	Value
# Cells	1
Cell Volume, V	$(2.0 \times 3.0 \times 2.0) \cdot 10^7 = 1.2\text{e}+22 \text{ cm}^3$
Spatial Discretization	PWLD (8 Elements)
Simulation Time, T	6 weeks = 56 days
Initial Density, N_0	5.0 [$\frac{\text{atom}}{\text{b}-\text{cm}}$]
$\sigma_{s,1 \rightarrow 1}, \sigma_{s,1 \rightarrow 2}, \sigma_{s,2 \rightarrow 1}, \sigma_{s,2 \rightarrow 2}$	0.6, 0.2, 0.05, 1.4 [b]
$\sigma_{t,1}, \sigma_{t,2}, \sigma_{f,2}$	2.0, 5.3, 3.38 [b]
χ_1, χ_2, ν	0.85, 0.15, 2.23
E_f, \mathcal{P}_0	206.0 MeV, 60.0 W/cm ³
$t_{1/2}$ half-life	5 weeks = 35 days
Fixed-source solver tolerance	1.0e-07
Eigenvalue, eigenvector tolerance	1.0e-07, 1.0e-06

We solved this problem using each of the four time stepping schemes with time-steps equal to 1, 2, 4, and 8 days. For the explicit and implicit Euler schemes, we ran with both 1 and 10 Bateman sub-cycles per time step. After each run, we compared the numerical QOI to the analytic QOI, and compiled the results to compute the convergence rate of each scheme.

The results are given in Fig. 5.1. The figure shows that the explicit Euler and implicit Euler schemes are converging at a first order rate, as expected. Moreover, we find that the schemes are more accurate when more Bateman sub-cycles are used (in this case by about an order of magnitude). The modified Euler and Runge-

Kutta 4 schemes appear to achieve their expected convergence rates of 2.0, and 4.0, respectively.

Another observation is that the schemes with higher convergence rates are more accurate, even when normalized per solution to the eigenvalue equations. For example, the explicit and modified Euler schemes require 1 and 2 k-solves per time step, respectively. Therefore, the explicit Euler scheme performs the same number of k-solves with the time step of 1.0 days as the modified Euler scheme performs with a time step of 2.0 days. The 2.0 day modified Euler result, however, is two orders of magnitude more accurate than the 1.0 day explicit Euler result, at least for this problem. This might suggest that a more accurate method with longer time steps is preferable than a lower accuracy method with more time steps.

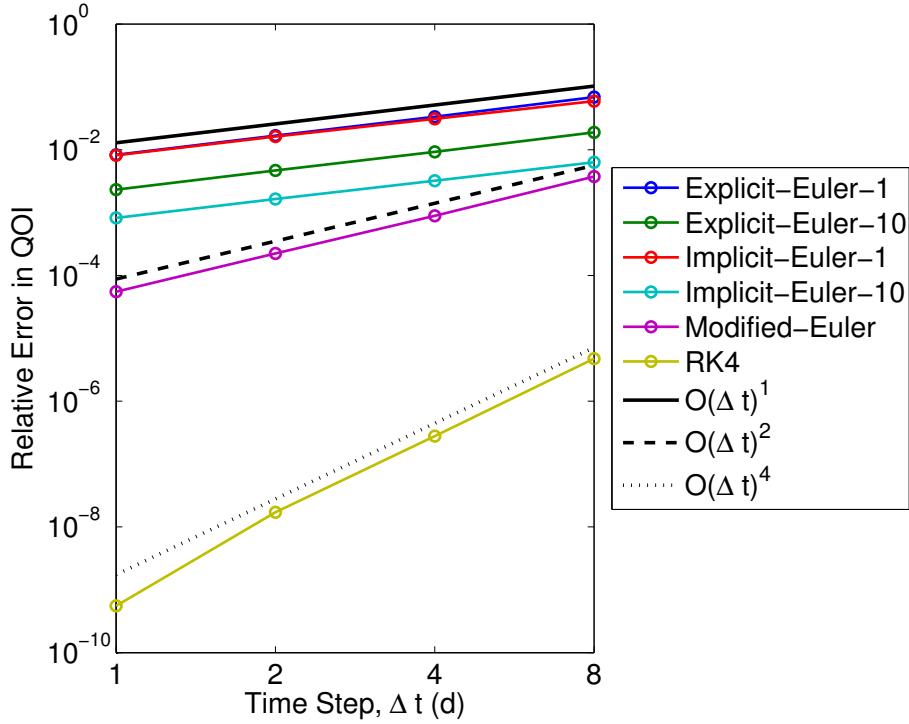
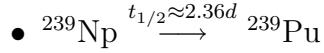


Figure 5.1: Convergence rates of the Runge-Kutta schemes for infinite medium problem. The explicit Euler, implicit Euler, modified Euler, and Runge-Kutta 4 schemes are converging at the expected rates of 1.0, 1.0, 2.0, and 4.0. Note: the number next to the explicit and implicit Euler schemes indicates the number of Bateman sub-cycles.

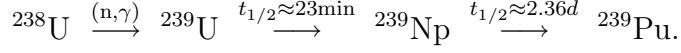
5.1.2 A 33 group model of the ^{239}Pu production chain in an infinite medium

In this problem, we simulate the production of Plutonium-239 in an infinite medium using the 33-group ANL cross sections (see App. F.2). We track 6 nuclides: Uranium-235, Uranium-238, Neptunium-239, Plutonium-239, MU-35, and MU-38 (lumped fission products). The depletion chain is as follows:

- $^{235}\text{U} \xrightarrow{(\text{n},\text{f})} \text{MU-35}$
- $^{238}\text{U} \xrightarrow{(\text{n},\text{f})} \text{MU-38}$
- $^{238}\text{U} \xrightarrow{(\text{n},\gamma)} {}^{239}\text{Np}$



Note that the true Plutonium-239 production chain includes the following reaction chain:



The half-life of Uranium-239, however, is short enough to assume that it immediately decays into Neptunium-239. We therefore do not include Uranium-239, which allows us to take longer time steps. The initial densities and other problem parameters are given in Table 5.2.

Table 5.2: Problem definition for the 33-group Plutonium-239 production test problem

Parameter	Value
# Cells	1
Cell Volume, V	$(2.0 \times 3.0 \times 2.0) \cdot 10^7 = 1.2\text{e+}22 \text{ cm}^3$
Spatial Discretization	PWLD (8 Elements)
Simulation Time, T	16 days
Initial Densities: ^{235}U , ^{238}U , ^{239}Np	0.4, 0.3, 0.0 [$\frac{\text{atom}}{\text{b-cm}}$]
Initial Densities: ^{239}Pu , MU-35, MU-38	0.0, 3.6, 3.9 [$\frac{\text{atom}}{\text{b-cm}}$]
Initial Eigenvalue, $k_{\text{eff}}(t_0)$	0.63
Fixed Source, S_g	$1.8\text{e}15 \left[\frac{\text{n}}{\text{cm}^3\text{-s-MeV}} \right]$
Fixed-source solver tolerance	1.0e-07

The problem was solved using three different explicit Runge-Kutta methods: the explicit Euler method, with time steps of 0.5, 1.0, 2.0, and 4.0 days; the modified

Euler method, with time steps of 0.5, 1.0, 2.0, and 4.0 days, and the Runge-Kutta 4 method, with time steps of 2.0, 4.0, 8.0, and 16.0 days. The QOIs from the problem were simply the number densities for each of the six nuclides at the end of the simulation.

We then generated a reference solution using the ODE solver suite in Matlab[55]. This tool uses embedded Runge-Kutta schemes to perform dynamic error control and stability analysis to solve a user-defined set of ODEs to within user-defined absolute and relative tolerances. The dynamic equations must be of the form

$$y' = f(t, y),$$

where y is a vector of unknowns and the function $f(t, y)$ returns a vector of time derivative of those unknowns. Specifically, our equations are

$$\begin{aligned} \frac{dN_i}{dt} &= B(N, \phi), \quad i = 1 \dots 6 \\ \frac{d\phi_g}{dt} &= D\phi - S_g, \quad g = 1 \dots 33 \end{aligned}$$

where D is a diffusion matrix with entries

$$D_{kj} = \delta_{kj} \sum_i N_i \sigma_{t,i,k} - \sum_i N_i \sigma_{i,s,j \rightarrow k} - \chi_k \sum_i N_i \nu_{i,j} \sigma_{f,i,j}.$$

Here χ is the density weighted spectrum used in PDT

$$\chi_k = \frac{\sum_i N_i \chi_{i,k}}{\sum_i N_i}, \quad N_i > 0.$$

The diffusion approximation is valid here because it is an infinite medium; thus the PDT transport solution should be equivalent to the Matlab diffusion solution.

Finally, for the initial flux conditions, ϕ_0 , we simply solve a steady-state diffusion problem

$$D_0\phi_0 = S$$

where $D_0 = D(N_0)$.

The Matlab solution was integrated with absolute and relative tolerances of $1.0e - 3$ and $1.0e - 12$, respectively. The resulting nuclide densities, which we will call the reference solution, were found to be in close agreement with the PDT results. Table 5.3 compares the reference solution, least accurate PDT solution (explicit Euler with time step 4.0 days), and most accurate PDT solution (RK4 with time step 2.0 days).

Table 5.3: Reference and PDT terminal densities for the 33g ^{239}Pu production problem (all densities in units of $\left[\frac{\text{atom}}{\text{b}-\text{cm}}\right]$)

Nuclide	Reference Density	Explicit Euler 4.0 days	Runge-Kutta 4 2.0 days
^{235}U	3.60665e-01	3.59907e-01	3.60665e-01
^{238}U	2.96605e-01	2.96575e-01	2.96605e-01
^{239}Np	5.84058e-04	5.87053e-04	5.84013e-04
^{239}Pu	2.15729e-03	2.21340e-03	2.15733e-03
MU-35	3.58154e+00	3.58190e+00	3.58154e+00
MU-38	3.83109e+00	3.83065e+00	3.83109e+00

Figures 5.2, 5.3, and 5.4 show the relative error between the reference and PDT solutions as a function of time step for the explicit Euler, modified Euler, and Runge-

Kutta 4 schemes, respectively. Note that there are 6 lines in each plot; the relative errors for Uranium-238 and MU-38 are nearly identical in each case. Again, we see that the schemes approach the reference solution at a rate of (Δt) , (Δt^2) , and (Δt^4) , as predicted by theory.

These results provide further confidence that (a) the Runge-Kutta schemes are implemented correctly, and (b) the process of matching nuclide parent/child pairs for reaction and decays in PDT is consistent with the hand-coded parent/child pairs that generated the reference solution.

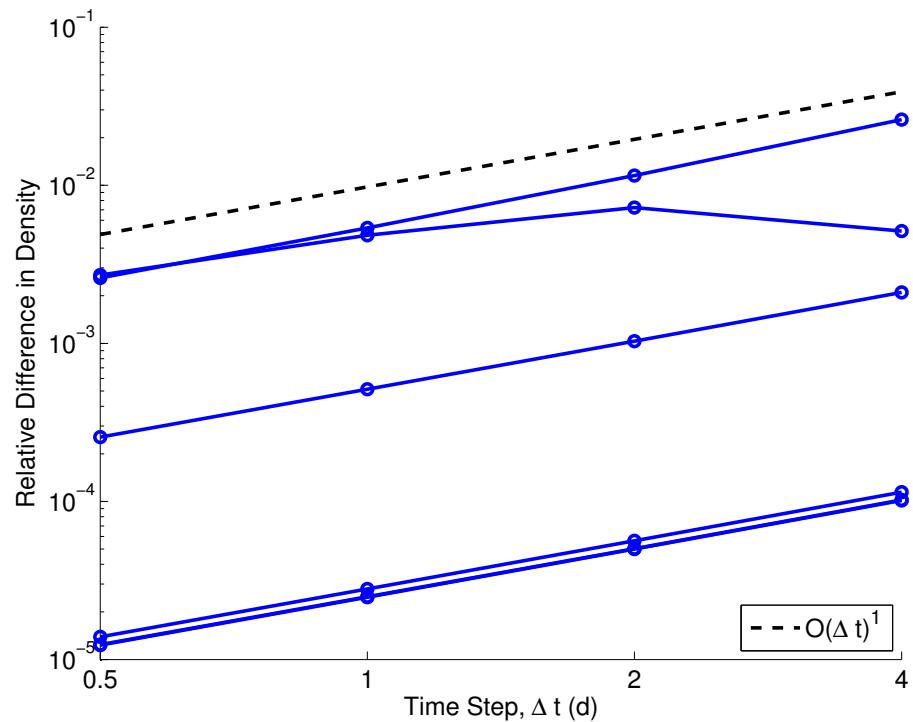


Figure 5.2: Convergence rate of the explicit Euler scheme (to the Matlab reference solution). The scheme appears to achieve the expected first order rate. Note: there are 6 lines drawn; 2 are overlaid.

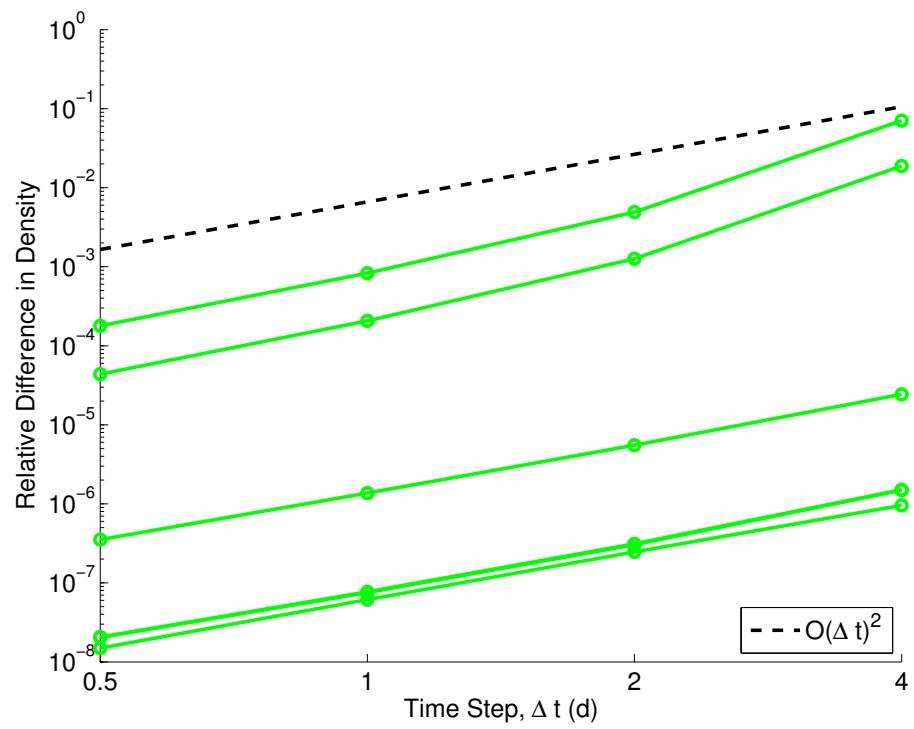


Figure 5.3: Convergence rate of the modified Euler scheme (to the Matlab reference solution). The scheme appears to achieve the expected second order rate. Note: there are 6 lines drawn; 2 are overlaid

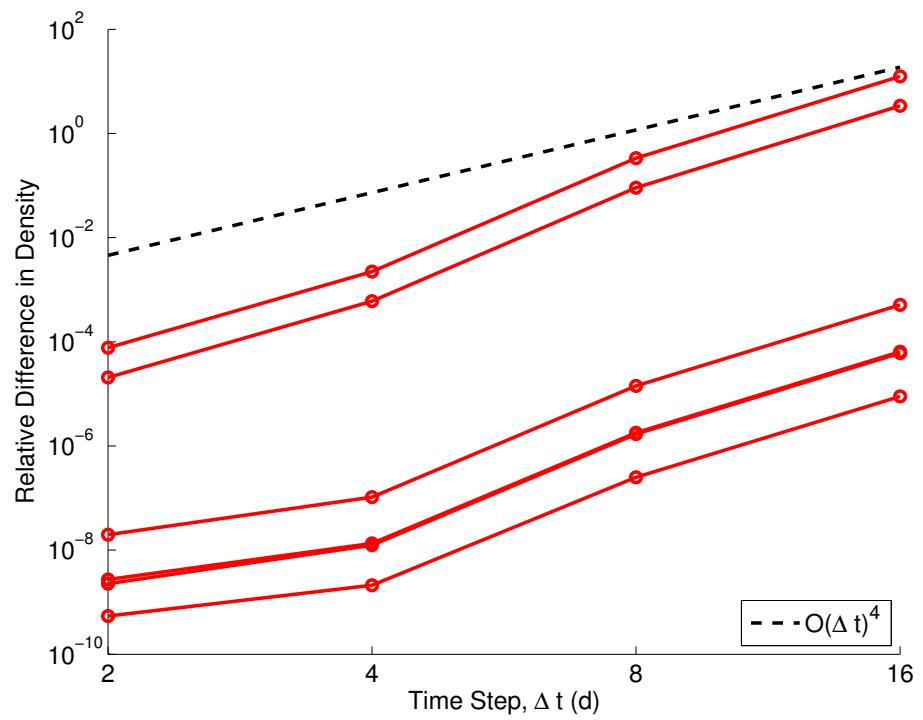


Figure 5.4: Convergence rate of the Runge-Kutta 4 scheme (to the Matlab reference solution). The scheme appears to achieve the expected fourth order rate. Note: there are 6 lines drawn; 2 are overlaid

5.2 Verification Problems Targeting the Steady-State Adjoint Transport Solver

In this section we describe verification problems that are designed to test the adjoint transport solver in PDT. They are steady-state problems (no depletion), and we seek to verify that the sensitivity estimates produced with the adjoint formalism are accurate and converging at the proper rate.

5.2.1 Infinite medium problem with analytic sensitivities

In the limit of an infinite medium, we can compute analytic sensitivities of a QOI and compare those to the numerical sensitivities produced by PDT. In an infinite medium, the diffusion equation with no leakage becomes valid. We model an infinite medium in PDT using a *really* big single cell problem.

The problem has two components, cmp1 and cmpD, with densities 0.8 and 2.1 [$\frac{\text{atom}}{\text{b-sec}}$], respectively, with cross sections given in Table 5.4.

Table 5.4: Synthetic cross sections for cmp1 and cmpD used in the steady-state, infinite medium verification problem

Cross Section [b]	cmp1	cmpD
$\sigma_{t,1}$	2.3	4.0
$\sigma_{t,2}$	1.4	1.0
$\sigma_{s,1 \rightarrow 1}$	0.35	0.5
$\sigma_{s,1 \rightarrow 2}$	0.52	0.33
$\sigma_{s,2 \rightarrow 1}$	0.29	0.18
$\sigma_{s,2 \rightarrow 2}$	0.15	0.5

The diffusion equations for group 1 and group 2 are

$$\Sigma_{t,1}\phi_1 = \Sigma_{s,1\rightarrow 1}\phi_1 + \Sigma_{s,2\rightarrow 1}\phi_2 + S_1$$

$$\Sigma_{t,2}\phi_2 = \Sigma_{s,1\rightarrow 2}\phi_1 + \Sigma_{s,2\rightarrow 2}\phi_2 + S_2$$

where

$$\Sigma_{t,g} = N_D\sigma_{t,D,g} + N_1\sigma_{t,1,g}$$

$$\Sigma_{s,g\rightarrow g'} = N_D\sigma_{s,D,g\rightarrow g'} + N_1\sigma_{s,1,g\rightarrow g'}$$

and $S_1=4200$ and $S_2=1500$ $\left[\frac{n}{cm^3\text{-sec-MeV}}\right]$ are isotropic, fixed neutron sources. The analytic solutions to these equations are

$$\begin{aligned}\phi_1 &= \frac{S_2\Sigma_{s,2\rightarrow 1} + S_1(\Sigma_{t,2} - \Sigma_{s,2\rightarrow 2})}{\Sigma_{t,2}\Sigma_{t,1} + \Sigma_{s,2\rightarrow 2}\Sigma_{s,1\rightarrow 1} - \Sigma_{t,2}\Sigma_{s,1\rightarrow 1} - \Sigma_{s,2\rightarrow 2}\Sigma_{t,1} - \Sigma_{s,1\rightarrow 2}\Sigma_{s,2\rightarrow 1}} \\ \phi_2 &= \frac{\Sigma_{t,1}\phi_1 - \Sigma_{s,1\rightarrow 1}\phi_1 - S_1}{\Sigma_{s,2\rightarrow 1}}.\end{aligned}$$

Now we define our QOI as the total interaction rate with cmpD in the infinite medium (in the case of PDT, we must include a volume integral). The discrete relation for our QOI is

$$\mathcal{R} = N_D V \sum_g \sigma_{t,D,g} \sum_q \psi_{q,g} w_q = N_D V \sum_g \sigma_{t,D,g} \phi_g$$

and we wish to use the adjoint to compute $\frac{d\mathcal{R}}{d\sigma_j}$ for each σ_j in the problem. Analytic

derivatives are straightforward to compute:

$$\frac{d\mathcal{R}}{d\sigma_j} = N_D V \left[\phi_1 \frac{\partial \sigma_{t,D,1}}{\partial \sigma_j} + \sigma_{t,D,1} \frac{\partial \phi_1}{\partial \sigma_j} + \phi_1 \frac{\partial \sigma_{t,D,2}}{\partial \sigma_j} + \sigma_{t,D,2} \left(\frac{\partial \phi_2}{\partial \phi_1} \frac{\partial \phi_1}{\partial \sigma_j} + \frac{\partial \phi_2}{\partial \sigma_j} \right) \right]$$

Table 5.5 gives the numerical sensitivity, analytic sensitivity, and relative difference between them. We find that the numerical sensitivities are accurate to the order of the iterative solver tolerances.

Table 5.5: The first derivative adjoint estimates from PDT agree with the analytic derivatives to roughly the iterative solver tolerance.

Cross Section	Adjoint Derivative	Analytic Derivative	Relative Difference
cmp1 $\sigma_{t,1}$	-3.851984e+27	-3.851984e+27	-3.365596e-08
cmp1 $\sigma_{t,2}$	-8.886579e+27	-8.886579e+27	-3.887295e-08
cmpD $\sigma_{t,1}$	-1.013732e+27	-1.013732e+27	-3.219049e-07
cmpD $\sigma_{t,2}$	-6.112938e+27	-6.112939e+27	-1.366759e-07
cmp1 $\sigma_{s,1 \rightarrow 1}$	3.851984e+27	3.851984e+27	-3.365596e-08
cmp1 $\sigma_{s,2 \rightarrow 1}$	7.288560e+27	7.288561e+27	-3.628653e-08
cmp1 $\sigma_{s,1 \rightarrow 2}$	4.696532e+27	4.696532e+27	-3.603594e-08
cmp1 $\sigma_{s,2 \rightarrow 2}$	8.886579e+27	8.886579e+27	-3.887295e-08
cmpD $\sigma_{s,1 \rightarrow 1}$	1.011146e+28	1.011146e+28	-3.369304e-08
cmpD $\sigma_{s,2 \rightarrow 1}$	1.913247e+28	1.913247e+28	-3.605133e-08
cmpD $\sigma_{s,1 \rightarrow 2}$	1.232840e+28	1.232840e+28	-3.616775e-08
cmpD $\sigma_{s,2 \rightarrow 2}$	2.332727e+28	2.332727e+28	-3.882472e-08

5.2.2 Source-driven, 33g detector response problem

In this problem, we use the adjoint formalism to produce parameter sensitivity estimates for a detector-response QOI and test those sensitivities against those generated via brute-force finite difference. The problem geometry is pictured in Fig. 5.5. The problem consists of a $3 \times 3 \times 3$ cell matrix. The inner-most cell contains detector material, and the outside 8 cells contain filler material. The detector material is a homogeneous mixture of Uranium-235 and MU-35 with number densities 1.3 and 2.0 [$\frac{\text{atom}}{\text{b}-\text{cm}}$], respectively. The filler material is a homogeneous mixture of Uranium-238, MU-35, and MU-38 with number densities 0.5, 1.0, and 1.0 [$\frac{\text{atom}}{\text{b}-\text{cm}}$], respectively. See Appendix F.2 for details on these materials.

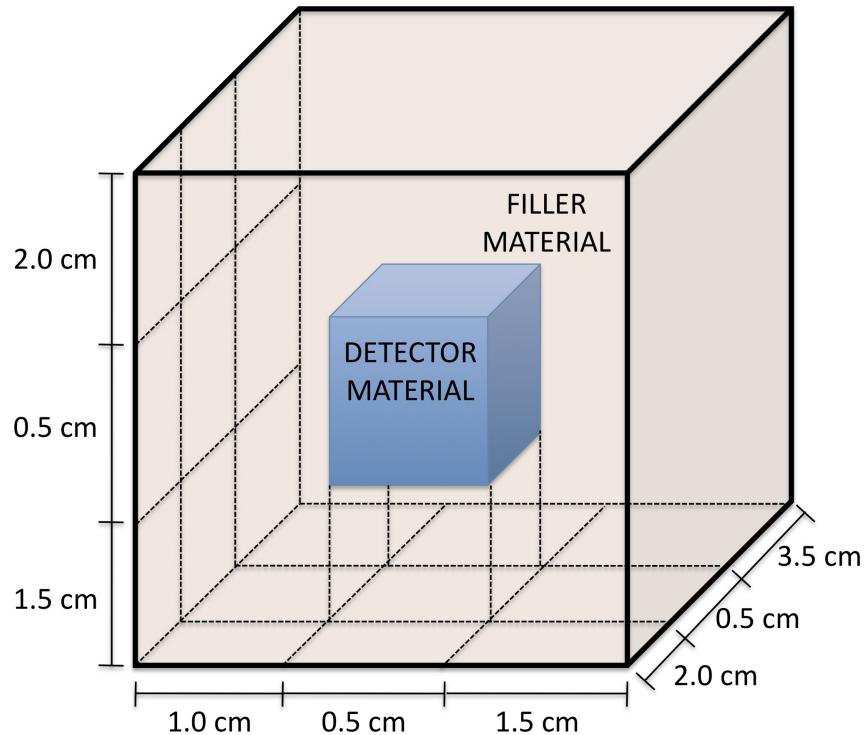


Figure 5.5: Geometry of the 33 group steady-state adjoint verification problem.

The 33-group problem has a fixed source of $4.2\text{e}+09 \left[\frac{\text{n}}{\text{cm}^3\text{-s-MeV}} \right]$ in each group. The k-eigenvalue of the system is around 0.86. We define a QOI as the total interaction rate in the detector volume, V_D ,

$$Q = \int dV_D \int dE \int d\Omega N_{235}(r) \sigma_{t,235}(E) \psi(r, E, \Omega),$$

which simulates a detector response. We will compute this QOI for the unperturbed cross sections, and we will use the adjoint formalism to compute its sensitivity with respect to the following group-wise data for each component: σ_t , σ_s , σ_f , and ν .

Our task in this verification problem is to test that these parameter sensitivities are indeed first order accurate with respect to the parameter perturbation. We test this by perturbing a single cross section, $\tilde{\sigma} = \sigma_0 + \Delta\sigma$ and re-running the forward problem to compute a perturbed QOI, \tilde{Q} . The perturbed QOI can be related to the nominal QOI as follows:

$$\tilde{Q} = Q + \Delta\sigma \frac{dQ}{d\sigma} + \mathcal{O}\left(\Delta\sigma\right)^2.$$

Therefore, if the derivative $\frac{dQ}{d\sigma}$ is correct, the error made in predicting the $\Delta Q = \tilde{Q} - Q$ by extrapolating the derivative should decrease proportional to the square of the parameter perturbation.

We ran a single adjoint solve to produce the full gradient of the QOI with respect to all of the parameters. Then, for each parameter, we ran 4 different perturbed problems with $\Delta\sigma = (0.001, 0.002, 0.004, \text{ and } 0.008)$ to compute 4 different values of ΔQ . For the scattering cross section, we only perturbed $\sigma_{s,g' \rightarrow g}$, $g = 1 \dots 33$, $g \leq g' \leq g + 3$ in order to cut down on the number of required runs.

Figures 5.6, 5.7, 5.8, and 5.9 show the error in predicting ΔQ as a function of

the perturbation in σ_t , σ_s , σ_f , and ν , respectively. Note that perturbed runs that resulted in a predicted *and* actual ΔQ less than the solver tolerance are not shown. Also, for obvious reasons, we only perturbed the fission data in Uranium-235 and Uranium-238.

Except for some noise where ΔQ was on the order of the solver tolerance, all convergence rates appear to be achieving second order; that is, when $\Delta\sigma$ is halved, the error in predicting ΔQ is cut by a factor of 4. This rate agrees with our expectations, indicating that the adjoint transport solver implemented in PDT is solving the adjoint equations correctly.

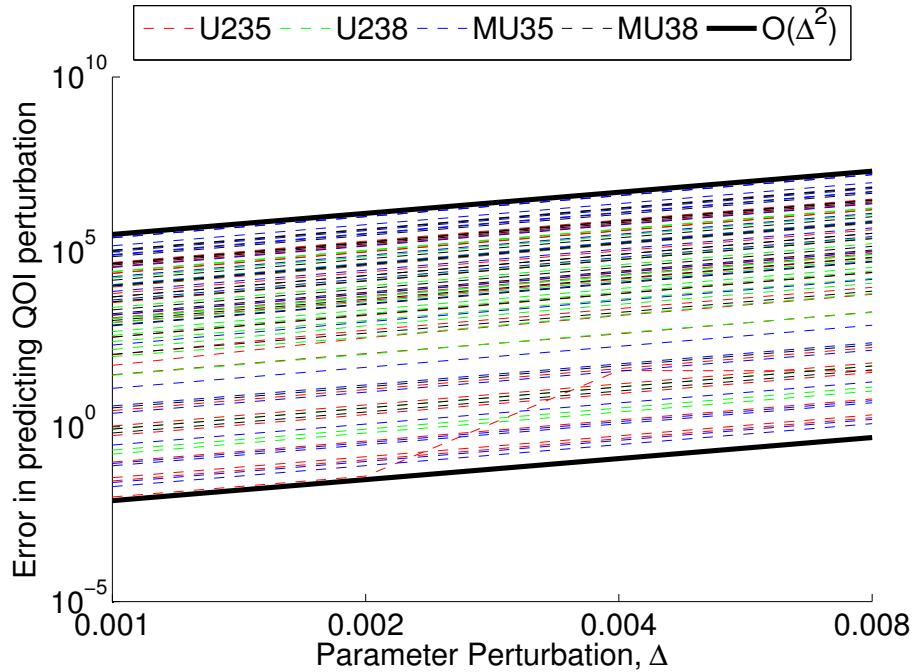


Figure 5.6: Convergence rates for $\frac{dQ}{d\sigma_t}$ in the 33 group steady-state adjoint transport problem. The error in predicting ΔQ as a function of the perturbation in $\Delta\sigma_t$ decreases as the square of the parameter perturbation.

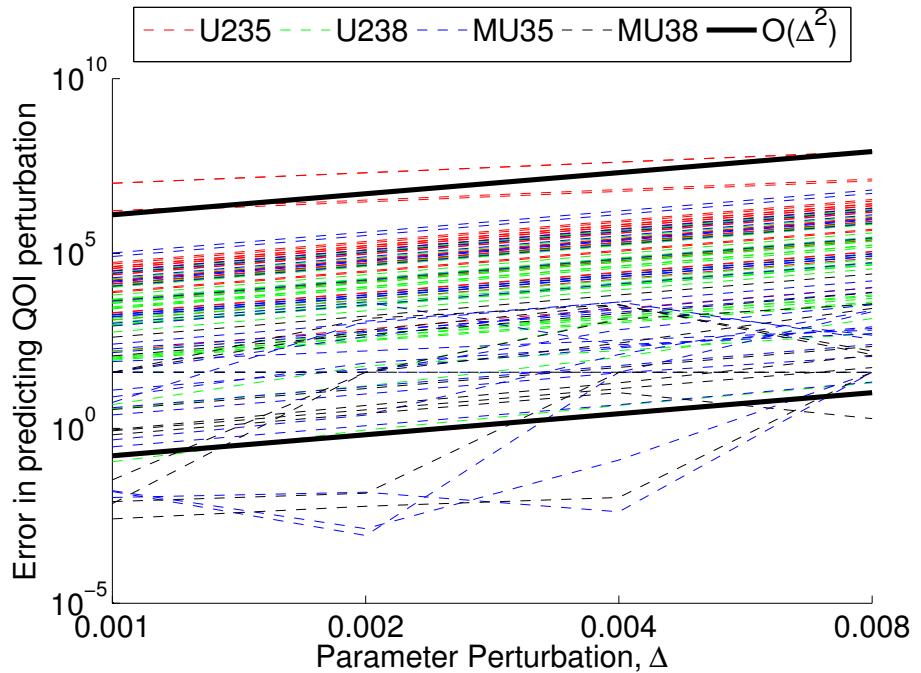


Figure 5.7: Convergence rates for $\frac{dQ}{d\sigma_s}$ in the 33 group steady-state adjoint transport problem. The error in predicting ΔQ as a function of the perturbation in $\Delta\sigma_s$ decreases as the square of the parameter perturbation.

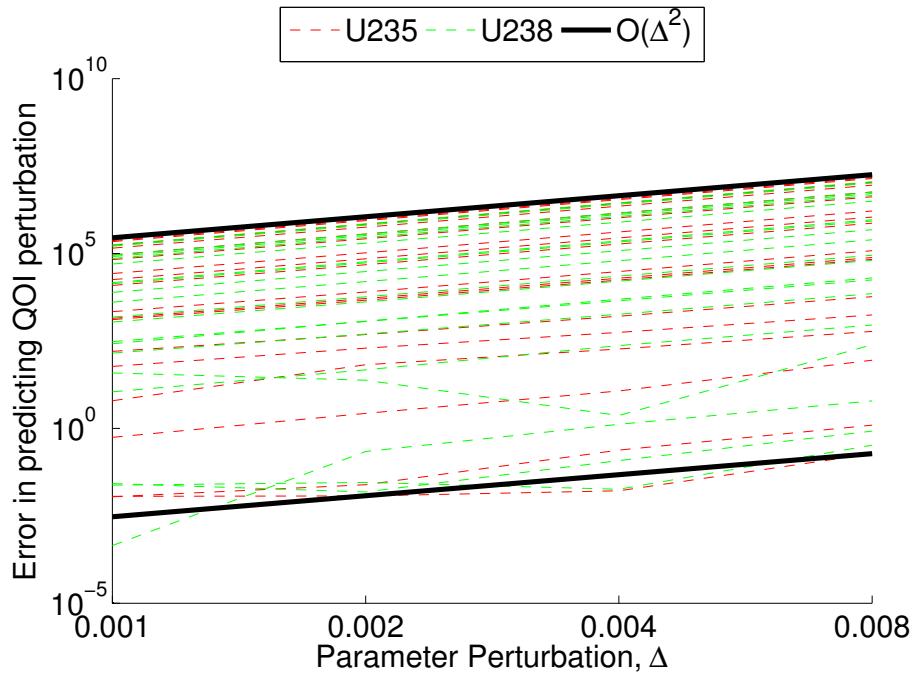


Figure 5.8: Convergence rates for $\frac{dQ}{d\sigma_f}$ in the 33 group steady-state adjoint transport problem. The error in predicting ΔQ as a function of the perturbation in $\Delta \sigma_f$ decreases as the square of the parameter perturbation.

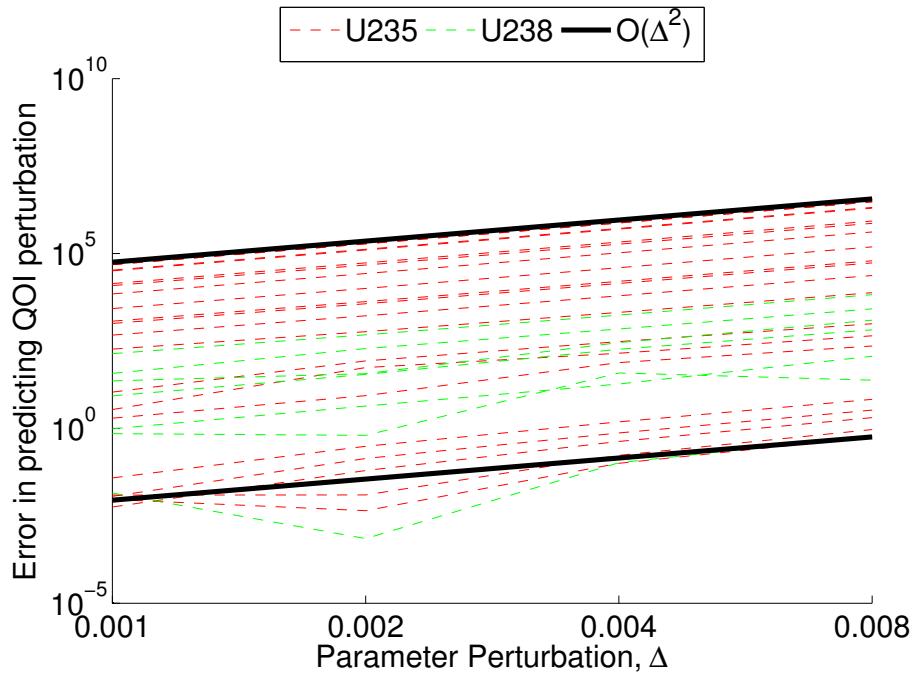


Figure 5.9: Convergence rates for $\frac{dQ}{d\nu}$ in the 33 group steady-state adjoint transport problem. The error in predicting ΔQ as a function of the perturbation in $\Delta \nu$ decreases as the square of the parameter perturbation.

5.3 A Two-Group, Infinite Medium k-Eigenvalue Problem Targeting the Full Depletion Perturbation Solver

This verification problem seeks to test our implementation of the depletion perturbation solver, which includes both the depletion and adjoint transport solvers that were addressed above. The purpose of this problem is two-fold. We first perform a walk-through of the discrete solution to a simple problem and demonstrate its Jacobian calculations, orthogonality properties, and sensitivity results. We then develop analytic, continuous solutions to the depletion problem and demonstrate convergence properties of the forward and adjoint integration as well as the adjoint-based error estimate.

5.3.1 A walk-through of the solution procedure

Here we develop analytic expressions for the discrete solution to an infinite medium k-eigenvalue verification problem and compare those expressions to numerical results from our implementation in PDT. This is a two group problem with a single component. The cross sections of the component are

$$\begin{aligned}\sigma_{t,1} &= \sigma_{R,1} + \sigma_{s,1 \rightarrow 1} + \sigma_{s,1 \rightarrow 2} \\ \sigma_{t,2} &= \sigma_{\gamma,2} + \sigma_{f,2} + \sigma_{s,2 \rightarrow 1} + \sigma_{s,2 \rightarrow 2} \\ &= \sigma_{R,2} + \sigma_{s,2 \rightarrow 1} + \sigma_{s,2 \rightarrow 2}\end{aligned}$$

The component emits E_f MeV per fission and ν neutrons per fission, with fraction χ_1 into group 1 and χ_2 into group 2. Finally, the infinite medium operates at a

constant power density \mathcal{P}_0 . The governing equations, Eqs. (3.54) – (3.57), are

$$\begin{aligned} \frac{dN}{dt} - \left\langle bN \right\rangle_{E,\mathcal{D},\Omega} &= 0 \\ \Omega \cdot \nabla \psi(E, \Omega) + \sigma_t(E)N\psi(E, \Omega) - \frac{1}{4\pi} \int dE' \sigma_{s,E' \rightarrow E} N \int d\Omega' \psi(E', \Omega') \\ &\quad - \lambda \frac{\chi(E)}{4\pi} \int dE' \int d\Omega' \nu N \sigma_f(E') \psi(E', \Omega') = 0 \\ \mathcal{P}_0 - \frac{A}{V} \left\langle E_f N \sigma_f \psi \right\rangle_{E,\mathcal{D},\Omega} &= 0 \\ N(0) = N_0, \quad t \in [0, T]. \end{aligned}$$

After accounting for the infinite medium geometry and two-group cross sections, our equations reduce to

$$\frac{dN}{dt} - \left\langle -\frac{AN\sigma_R\psi F_b^{cm}}{V} \right\rangle_{E,\mathcal{D},\Omega} = \frac{dN}{dt} + 4\pi ANF_b^{cm} [\sigma_{R,1}\psi_1 + \sigma_{R,2}\psi_2] = 0 \quad (5.3)$$

$$\sigma_{t,1}N\psi_1 - N(\psi_1\sigma_{s,1 \rightarrow 1} + \psi_2\sigma_{s,2 \rightarrow 1}) - \lambda\chi_1\nu N\sigma_{f,2}\psi_2 = 0 \quad (5.4)$$

$$\sigma_{t,2}N\psi_2 - N(\psi_1\sigma_{s,1 \rightarrow 2} + \psi_2\sigma_{s,2 \rightarrow 2}) - \lambda\chi_2\nu N\sigma_{f,2}\psi_1 = 0 \quad (5.5)$$

$$\mathcal{P}_0 - A4\pi N E_f \sigma_{f,2} \psi_2 = 0 \quad (5.6)$$

$$N(0) = N_0, \quad t \in [0, T], \quad (5.7)$$

where we have assumed due to the infinite medium that $\psi_{gq} = \psi_g$ and $\phi_g = 4\pi\psi_g$.

Equations (5.4) and (5.5) give the following eigenfunctions, respectively:

$$\psi_1 = \frac{\lambda\chi_1\nu\sigma_{f,2} + \sigma_{s,2 \rightarrow 1}}{\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}} \psi_2 \quad (5.8)$$

$$\psi_1 = \frac{\lambda\chi_2\nu\sigma_{f,2} + \sigma_{s,2 \rightarrow 2} - \sigma_{t,2}}{\sigma_{s,1 \rightarrow 2}} \psi_2 \quad (5.9)$$

Eigenfunction (5.8) is strictly positive, so it must be the fundamental mode. If we

substitute Eq. (5.8) for ψ_1 in Eq. (5.5) we find the fundamental eigenvalue

$$\lambda_0 = \frac{(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})(\sigma_{t,2} - \sigma_{s,2 \rightarrow 2}) - \sigma_{s,1 \rightarrow 2}\sigma_{s,2 \rightarrow 1}}{\nu\sigma_{f,2}[\chi_1\sigma_{s,1 \rightarrow 2} + \chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})]}$$

If we substitute Eq. (5.9) for ψ_1 into Eq. (5.5) we find

$$\lambda_1 = \frac{(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})(\sigma_{t,2} - \sigma_{s,2 \rightarrow 2}) + \sigma_{s,1 \rightarrow 2}\sigma_{s,2 \rightarrow 1}}{\nu\sigma_{f,2}[\chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}) - \chi_1\sigma_{s,1 \rightarrow 2}]}$$

Finally from the power constraint we have

$$A = \frac{\mathcal{P}_0}{4\pi N E_f \sigma_{f,2} \psi_2}. \quad (5.10)$$

Let's define an arbitrary value $\psi_{2,0}$ as the initial un-normalized value of the group-2 angular flux. Substituting our expression for λ_0 into Eq. (5.8) we find

$$\begin{aligned} \psi_{1,0} &= \frac{\lambda_0 \chi_1 \nu \sigma_{f,2} + \sigma_{s,2 \rightarrow 1}}{\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}} \psi_{2,0} \\ &= \frac{1}{\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}} \left[\frac{(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})(\sigma_{t,2} - \sigma_{s,2 \rightarrow 2}) - \sigma_{s,1 \rightarrow 2}\sigma_{s,2 \rightarrow 1}}{\nu\sigma_{f,2}[\chi_1\sigma_{s,1 \rightarrow 2} + \chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})]} \chi_1 \nu \sigma_{f,2} + \sigma_{s,2 \rightarrow 1} \right] \psi_{2,0} \\ &= \frac{1}{\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}} \left[\frac{\chi_1(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})(\sigma_{t,2} - \sigma_{s,2 \rightarrow 2}) - \chi_1\sigma_{s,1 \rightarrow 2}\sigma_{s,2 \rightarrow 1}}{\chi_1\sigma_{s,1 \rightarrow 2} + \chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})} + \sigma_{s,2 \rightarrow 1} \right] \psi_{2,0} \\ &= \frac{1}{\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}} \left[\frac{\chi_1(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})(\sigma_{t,2} - \sigma_{s,2 \rightarrow 2}) + \chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})\sigma_{s,2 \rightarrow 1}}{\chi_1\sigma_{s,1 \rightarrow 2} + \chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})} \right] \psi_{2,0} \\ &= \frac{\chi_1(\sigma_{t,2} - \sigma_{s,2 \rightarrow 2}) + \chi_2\sigma_{s,2 \rightarrow 1}}{\chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}) + \chi_1\sigma_{s,1 \rightarrow 2}} \psi_{2,0} \end{aligned}$$

Note that this eigenfunction can also be expressed in terms of the removal cross

sections:

$$\begin{aligned}\psi_{1,0} &= \frac{\chi_1(\sigma_{t,2} - \sigma_{s,2 \rightarrow 2}) + \chi_2 \sigma_{s,2 \rightarrow 1}}{\chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}) + \chi_1 \sigma_{s,1 \rightarrow 2}} \psi_{2,0} \\ &= \frac{\chi_1 \sigma_{R,2} + \sigma_{s,2 \rightarrow 1}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \psi_{2,0}\end{aligned}$$

Then our initial state is

$$\begin{aligned}N(0) &= N_0 \\ \psi_1(0) &= \frac{\chi_1(\sigma_{t,2} - \sigma_{s,2 \rightarrow 2}) + \chi_2 \sigma_{s,2 \rightarrow 1}}{\chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}) + \chi_1 \sigma_{s,1 \rightarrow 2}} \psi_{2,0} \\ \psi_2(0) &= \psi_{2,0} \\ A(0) &= \frac{\mathcal{P}_0}{4\pi N_0 E_f \sigma_{f,2} \psi_{2,0}}\end{aligned}$$

Now consider advancing the densities to $t = T$ with a single time-step using the explicit Euler method. The terminal density value is

$$\begin{aligned}N(T) &= N_0 + T \left\langle bN(0) \right\rangle_{E,\mathcal{D},\Omega} \\ &= N_0 - T 4\pi A(0) N_0 F_b^{\text{cm}} [\sigma_{R,1} \psi_1(0) + \sigma_{R,2} \psi_2(0)] \\ &= N_0 - T \frac{4\pi N_0 \mathcal{P}_0 F_b^{\text{cm}}}{4\pi N_0 E_f \sigma_{f,2} \psi_{2,0}} \left[\frac{\chi_1 \sigma_{R,2} + \sigma_{s,2 \rightarrow 1}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \sigma_{R,1} + \sigma_{R,2} \right] \psi_{2,0} \\ &= N_0 - \frac{T \mathcal{P}_0 F_b^{\text{cm}}}{E_f \sigma_{f,2}} \left[\frac{\chi_1 \sigma_{R,1} \sigma_{R,2} + \sigma_{R,1} \sigma_{s,2 \rightarrow 1} + \chi_2 \sigma_{R,1} \sigma_{R,2} + \sigma_{R,2} \sigma_{s,1 \rightarrow 2}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \right] \\ &= N_0 - \frac{T \mathcal{P}_0 F_b^{\text{cm}}}{E_f \sigma_{f,2}} \left[\frac{\sigma_{R,1} \sigma_{R,2} + \sigma_{R,1} \sigma_{s,2 \rightarrow 1} + \sigma_{R,2} \sigma_{s,1 \rightarrow 2}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \right] \\ &\equiv N_T,\end{aligned}\tag{5.11}$$

which can be verified to have units of $\left[\frac{\text{atom}}{\text{cm-b}}\right]$. If we again assume an arbitrary value

$\psi_{2,T}$ for $\psi_2(T)$, then the terminal forward state is

$$N(T) = N_T$$

$$\psi_1(T) = \frac{\chi_1(\sigma_{t,2} - \sigma_{s,2 \rightarrow 2}) + \chi_2 \sigma_{s,2 \rightarrow 1}}{\chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}) + \chi_1 \sigma_{s,1 \rightarrow 2}} \psi_{2,T}$$

$$\psi_2(T) = \psi_{2,T}$$

$$A(T) = \frac{\mathcal{P}_0}{4\pi N_T E_f \sigma_{f,2} \psi_{2,T}}$$

It is instructive to consider two different QOIs: one that does not depend on the flux, and one that does. We'll consider the former in detail, then return to the latter. Define the first QOI, Q , as the number of mols of the component present at $t = T$ in the (*really big*) medium. Keeping inline with our formalism, we express the QOI as written in Eq. (3.74)

$$Q = \int \frac{N(T)}{N_A} dV = \left\langle R(N) \right\rangle_{E,\mathcal{D},\Omega} = \left\langle \frac{1}{\langle \rangle_{E,\Omega}} \frac{N(T)}{N_A} \right\rangle_{E,\mathcal{D},\Omega} = \frac{VN(T)}{N_A}$$

where again N_A is Avogadro's number in units of $\left[\frac{\text{atom}}{\text{mol}} \frac{\text{cm}^2}{\text{b}} \right]$. Suppose we are interested in computing the derivative of this QOI with respect to each total cross section, each scattering cross section, the number of neutrons per fission, the fission cross section in group 2, and the initial condition N_0 . Using Eq. (5.11), we find the

derivatives of the discrete equations are as follows:

$$\frac{dQ}{d\sigma_{t,1}} = -\frac{VT\mathcal{P}_0F_b^{cm}}{E_f\sigma_{f,2}N_A} \left[\frac{\sigma_{R,2}\sigma_{s,1\rightarrow 2}\chi_1 + \sigma_{s,1\rightarrow 2}\sigma_{s,2\rightarrow 1}}{[\chi_2\sigma_{R,1} + \sigma_{s,1\rightarrow 2}]^2} \right] \quad (5.12)$$

$$\frac{dQ}{d\sigma_{t,2}} = -\frac{VT\mathcal{P}_0F_b^{cm}}{E_f\sigma_{f,2}N_A} \left[\frac{\sigma_{R,1} + \sigma_{s,1\rightarrow 2}}{\chi_2\sigma_{R,1} + \sigma_{s,1\rightarrow 2}} \right] \quad (5.13)$$

$$\frac{dQ}{d\sigma_{s,1\rightarrow 1}} = -\frac{VT\mathcal{P}_0F_b^{cm}}{E_f\sigma_{f,2}N_A} \left[\frac{\chi_2\sigma_{R,1}(\chi_1\sigma_{R,2} + \sigma_{s,2\rightarrow 1})}{[\chi_2\sigma_{R,1} + \sigma_{s,1\rightarrow 2}]^2} \right] \quad (5.14)$$

$$\frac{dQ}{d\sigma_{s,1\rightarrow 2}} = \frac{VT\mathcal{P}_0F_b^{cm}}{E_f\sigma_{f,2}N_A} \left[\frac{\chi_1\sigma_{R,1}(\chi_1\sigma_{R,2} + \sigma_{s,2\rightarrow 1})}{[\chi_2\sigma_{R,1} + \sigma_{s,1\rightarrow 2}]^2} \right] \quad (5.15)$$

$$\frac{dQ}{d\sigma_{s,2\rightarrow 1}} = -\frac{VT\mathcal{P}_0F_b^{cm}}{E_f\sigma_{f,2}N_A} \left[\frac{\chi_2\sigma_{R,1}}{\chi_2\sigma_{R,1} + \sigma_{s,1\rightarrow 2}} \right] \quad (5.16)$$

$$\frac{dQ}{d\sigma_{s,2\rightarrow 2}} = \frac{VT\mathcal{P}_0F_b^{cm}}{E_f\sigma_{f,2}N_A} \left[\frac{\chi_1\sigma_{R,1}}{\chi_2\sigma_{R,1} + \sigma_{s,1\rightarrow 2}} \right] \quad (5.17)$$

$$\frac{dQ}{d\nu} = 0 \quad (5.18)$$

$$\frac{dQ}{d\sigma_{f,2}} = \frac{VT\mathcal{P}_0F_b^{cm}}{E_f\sigma_{f,2}^2N_A} \left[\frac{\sigma_{R,1}\sigma_{R,2} + \sigma_{R,1}\sigma_{s,2\rightarrow 1} + \sigma_{R,2}\sigma_{s,1\rightarrow 2}}{\chi_2\sigma_{R,1} + \sigma_{s,1\rightarrow 2}} \right] \quad (5.19)$$

$$\frac{dQ}{dN_0} = \frac{V}{N_A} \quad (5.20)$$

Our task is to develop closed-form expressions for these derivatives using our adjoint methodology. We begin by noting that $\hat{A}^\dagger = 0$ and $\hat{\psi}^\dagger = 0$ by Eqs. (3.82) and (3.83), respectively. Thus, using Eq. (3.84), the terminal condition for the adjoint density is

$$N^\dagger(T) = \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} = \frac{V}{N_A}$$

which has units of $\left[\frac{\text{b}-\text{cm}-\text{mol}}{\text{atom}} \right]$. Now via Eq. (3.78), we can compute the terminal

value of the adjoint power normalization constant, $A^\dagger(T)$:

$$A^\dagger(T) \frac{\langle \Sigma_E \psi \rangle_{E,\mathcal{D},\Omega}}{V} = \left\langle N^\dagger \frac{\partial bN}{\partial A}(T) \right\rangle_{E,\mathcal{D},\Omega}$$

If we break this down term-by-term, we get

$$\langle \Sigma_E \psi \rangle_{E,\mathcal{D},\Omega} = \int dV \int dE \int d\Omega N E_f \sigma_f \psi = 4\pi V N E_f \sigma_{f,2} \psi_2,$$

which has units of $\left[\frac{\text{MeV}}{\text{s}}\right]$, and

$$\begin{aligned} \left\langle N^\dagger \frac{\partial bN}{\partial A}(T) \right\rangle_{E,\mathcal{D},\Omega} &= \left\langle \frac{V}{N_A} \frac{\partial}{\partial A} \left[-\frac{AN\sigma_R \psi F_b^{\text{cm}}}{V} \right]_{t=T} \right\rangle_{E,\mathcal{D},\Omega} \\ &= - \left\langle \frac{V}{N_A} \frac{N_T \sigma_R \psi(T) F_b^{\text{cm}}}{V} \right\rangle_{E,\mathcal{D},\Omega} \\ &= - \frac{4\pi V N_T F_b^{\text{cm}}}{N_A} \left[\sigma_{R,1} \psi_{1,T} + \sigma_{R,2} \psi_{2,T} \right] \\ &= - \frac{4\pi V N_T F_b^{\text{cm}} \psi_{2,T}}{N_A} \left[\frac{\sigma_{R,1} \sigma_{R,2} + \sigma_{R,1} \sigma_{s,2 \rightarrow 1} + \sigma_{R,2} \sigma_{s,1 \rightarrow 2}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \right] \end{aligned}$$

which has units of $\left[\frac{\text{mol}}{\text{s}}\right]$. Solving for $A^\dagger(T)$, we find

$$A^\dagger(T) = - \frac{V F_b^{\text{cm}}}{N_A \sigma_{f,2} E_f} \left[\frac{\sigma_{R,1} \sigma_{R,2} + \sigma_{R,1} \sigma_{s,2 \rightarrow 1} + \sigma_{R,2} \sigma_{s,1 \rightarrow 2}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \right]$$

with units of $\left[\frac{\text{mol-cm}^3}{\text{MeV}}\right]$. We now turn to the adjoint transport equation to compute $\psi_1^\dagger(T)$ and $\psi_2^\dagger(T)$. From Eq. (3.76), the applicable adjoint transport equation is

$$H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger = N^\dagger \frac{\partial bN}{\partial \psi} - A^\dagger \frac{A}{V_R} \Sigma_E \equiv S^\dagger. \quad (5.21)$$

We know the left-hand side of this equation is singular. We show this explicitly by

writing the $g = 1$ adjoint transport equation, which gives an expression for ψ_2^\dagger in terms of ψ_1^\dagger :

$$\begin{aligned} \sigma_{t,1}N\psi_1^\dagger - N(\psi_1^\dagger\sigma_{s,1\rightarrow 1} + \psi_2^\dagger\sigma_{s,1\rightarrow 2}) - \lambda\nu\sigma_{f,1}N(\psi_1^\dagger\chi_2 + \psi_2^\dagger\chi_2) &= S_1^\dagger \\ [\sigma_{t,1}N - \sigma_{s,1\rightarrow 1}N]\psi_1^\dagger + [-\sigma_{s,1\rightarrow 2}N]\psi_2^\dagger &= S_1^\dagger \\ \rightarrow \psi_2^\dagger &= \frac{\sigma_{t,1} - \sigma_{s,1\rightarrow 1}}{\sigma_{s,1\rightarrow 2}}\psi_1^\dagger - \frac{S_1^\dagger}{\sigma_{s,1\rightarrow 2}N} \end{aligned} \quad (5.22)$$

We next write the $g = 2$ adjoint transport equation

$$\begin{aligned} \sigma_{t,2}N\psi_2^\dagger - N(\psi_1^\dagger\sigma_{s,2\rightarrow 1} + \psi_2^\dagger\sigma_{s,2\rightarrow 2}) - \lambda\nu\sigma_{f,2}N(\psi_1^\dagger\chi_1 + \psi_2^\dagger\chi_2) &= S_2^\dagger, \\ [-\sigma_{s,2\rightarrow 1}N - \lambda\nu\sigma_{f,2}N\chi_1]\psi_1^\dagger + [\sigma_{t,2}N - \sigma_{s,2\rightarrow 2}N - \lambda\nu\sigma_{f,2}N\chi_2]\psi_2^\dagger &= S_2^\dagger, \end{aligned}$$

and substitute Eq. (5.22) for ψ_2^\dagger :

$$\begin{aligned} &[-\sigma_{s,2\rightarrow 1}N - \lambda\nu\sigma_{f,2}N\chi_1]\psi_1^\dagger \\ &+ [\sigma_{t,2}N - \sigma_{s,2\rightarrow 2}N - \lambda\nu\sigma_{f,2}N\chi_2]\left[\frac{\sigma_{t,1} - \sigma_{s,1\rightarrow 1}}{\sigma_{s,1\rightarrow 2}}\psi_1^\dagger - \frac{S_1^\dagger}{\sigma_{s,1\rightarrow 2}N}\right] = S_2^\dagger, \\ &\left[\sigma_{s,1\rightarrow 2}\left(-\sigma_{s,2\rightarrow 1}N - \lambda\nu\sigma_{f,2}N\chi_1\right) + \left(\sigma_{t,2}N - \sigma_{s,2\rightarrow 2}N - \lambda\nu\sigma_{f,2}N\chi_2\right)\left(\sigma_{t,1} - \sigma_{s,1\rightarrow 1}\right)\right]\psi_1^\dagger \\ &= S_2^\dagger\sigma_{s,1\rightarrow 2} + S_1^\dagger\left[\sigma_{t,2}N - \sigma_{s,2\rightarrow 2}N - \lambda\nu\sigma_{f,2}N\chi_2\right]. \end{aligned}$$

Through a series of straightforward manipulations and a substitution of the expres-

sion for λ_0 , we find that the coefficient that multiplies ψ_1^\dagger is zero:

$$\begin{aligned}
& \sigma_{s,1 \rightarrow 2} \left(-\sigma_{s,2 \rightarrow 1} N - \lambda \nu \sigma_{f,2} N \chi_1 \right) \\
& + \left(\sigma_{t,2} N - \sigma_{s,2 \rightarrow 2} N - \lambda \nu \sigma_{f,2} N \chi_2 \right) \left(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1} \right) \\
= & N \left[\sigma_{t,1} \sigma_{t,2} - \sigma_{t,1} \sigma_{s,2 \rightarrow 2} - \sigma_{t,2} \sigma_{s,1 \rightarrow 1} + \sigma_{s,1 \rightarrow 1} \sigma_{s,2 \rightarrow 2} - \sigma_{s,2 \rightarrow 1} \sigma_{s,1 \rightarrow 2} \right. \\
& \quad \left. - \lambda \nu \sigma_{f,2} \left(\sigma_{s,1 \rightarrow 2} \chi_1 + \sigma_{t,1} \chi_2 - \sigma_{s,1 \rightarrow 1} \chi_2 \right) \right] \\
= & N \left[\sigma_{t,1} \sigma_{t,2} - \sigma_{t,1} \sigma_{s,2 \rightarrow 2} - \sigma_{t,2} \sigma_{s,1 \rightarrow 1} + \sigma_{s,1 \rightarrow 1} \sigma_{s,2 \rightarrow 2} - \sigma_{s,2 \rightarrow 1} \sigma_{s,1 \rightarrow 2} \right. \\
& \quad \left. - \left[(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}) (\sigma_{t,2} - \sigma_{s,2 \rightarrow 2}) - \sigma_{s,1 \rightarrow 2} \sigma_{s,2 \rightarrow 1} \right] \right] \\
= & 0.
\end{aligned}$$

Indeed the left hand side of Eq. (5.21) is singular. The Fredholm alternative theorem[50] states that a unique solution $\psi^\dagger(t)$ exists if and only if

$$\langle \psi(t), S^\dagger(t) \rangle_{E,\mathcal{D},\Omega} = 0.$$

In Sec. 3.3, we showed that this is satisfied if the QOI satisfies Eq. (3.73); for the terminal QOI, this condition is always met, so we should find a unique solution. To verify that the orthogonality condition is met, we expand each of the terms in S^\dagger :

$$\begin{aligned}
N^\dagger(T) \frac{\partial bN}{\partial \psi_g}(T) &= \frac{V}{N_A} \frac{\partial}{\partial \psi} \left[-\frac{A_T}{V} N_T \sigma_{R,g} \psi_g F_b^{\text{cm}} \right] = -\frac{\mathcal{P}_0 \sigma_{R,g} F_b^{\text{cm}}}{4\pi N_A E_f \sigma_{f,2} \psi_{2,T}} \\
A_T^\dagger \frac{A_T}{V} \Sigma_{E,g} &= -\frac{V F_b^{\text{cm}} \mathcal{P}_0 N_T \sigma_{f,g} E_f}{N_A \sigma_{f,2} E_f 4\pi N_T E_f \sigma_{f,2} \psi_{2,T} V} \left[\frac{\sigma_{R,1} \sigma_{R,2} + \sigma_{R,1} \sigma_{s,2 \rightarrow 1} + \sigma_{R,2} \sigma_{s,1 \rightarrow 2}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \right] \\
&= -\frac{F_b^{\text{cm}} \mathcal{P}_0 \sigma_{f,g}}{4\pi N_A E_f \sigma_{f,2}^2 \psi_{2,T}} \left[\frac{\sigma_{R,1} \sigma_{R,2} + \sigma_{R,1} \sigma_{s,2 \rightarrow 1} + \sigma_{R,2} \sigma_{s,1 \rightarrow 2}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \right]
\end{aligned}$$

The orthogonality condition requires that $S_{1,T}^\dagger \psi_{1,T} + S_{2,T}^\dagger \psi_{2,T} = 0$. We find:

$$\begin{aligned}
S_{1,T}^\dagger \psi_{1,T} &= -\frac{\mathcal{P}_0 \sigma_{R,1} F_b^{cm}}{4\pi E_f \sigma_{f,2} N_A \psi_{2,T}} \left[\frac{\chi_1 \sigma_{R,2} + \sigma_{s,2 \rightarrow 1}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \right] \psi_{2,T} \\
&= -\frac{\mathcal{P}_0 \sigma_{R,1} F_b^{cm} (\chi_1 \sigma_{R,2} + \sigma_{s,2 \rightarrow 1})}{4\pi E_f \sigma_{f,2} N_A (\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2})} \\
S_{2,T}^\dagger \psi_{2,T} &= \left[-\frac{\mathcal{P}_0 \sigma_{R,2} F_b^{cm}}{4\pi E_f \sigma_{f,2} \psi_{2,T} N_A} \right. \\
&\quad \left. + \frac{F_b^{cm} \mathcal{P}_0 \sigma_{f,2}}{4\pi E_f \sigma_{f,2}^2 \psi_{2,T} N_A} \left[\frac{\sigma_{R,1} \sigma_{R,2} + \sigma_{R,1} \sigma_{s,2 \rightarrow 1} + \sigma_{R,2} \sigma_{s,1 \rightarrow 2}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \right] \right] \psi_{2,T} \\
&= \frac{-\mathcal{P}_0 \sigma_{R,2} F_b^{cm} (\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2})}{4\pi E_f \sigma_{f,2} N_A (\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2})} \\
&\quad + \frac{\mathcal{P}_0 F_b^{cm} (\sigma_{R,1} \sigma_{R,2} + \sigma_{R,1} \sigma_{s,2 \rightarrow 1} + \sigma_{R,2} \sigma_{s,1 \rightarrow 2})}{4\pi E_f \sigma_{f,2} N_A (\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2})}
\end{aligned}$$

Summing the numerators of these two expressions, we find

$$\begin{aligned}
&- \mathcal{P}_0 \sigma_{R,1} F_b^{cm} (\chi_1 \sigma_{R,2} + \sigma_{s,2 \rightarrow 1}) - \mathcal{P}_0 \sigma_{R,2} F_b^{cm} (\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}) \\
&\quad + \mathcal{P}_0 F_b^{cm} (\sigma_{R,1} \sigma_{R,2} + \sigma_{R,1} \sigma_{s,2 \rightarrow 1} + \sigma_{R,2} \sigma_{s,1 \rightarrow 2}) \\
&= \mathcal{P}_0 F_b^{cm} \left[\sigma_{R,1} \sigma_{R,2} (-\chi_1 - \chi_2 + 1) - \sigma_{R,1} \sigma_{s,2 \rightarrow 1} + \sigma_{R,1} \sigma_{s,2 \rightarrow 1} \right. \\
&\quad \left. - \sigma_{R,2} \sigma_{s,1 \rightarrow 2} + \sigma_{R,2} \sigma_{s,1 \rightarrow 2} \right] \\
&= 0
\end{aligned}$$

Thus, the orthogonality condition is met and we should be able to solve uniquely for $\psi^\dagger(T)$. We first rewrite Eq. (5.22) with the form of S_2^\dagger :

$$\begin{aligned}
\psi_2^\dagger &= \frac{\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}}{\sigma_{s,1 \rightarrow 2}} \psi_1^\dagger - \frac{1}{N \sigma_{s,1 \rightarrow 2}} \left[-\frac{V}{N_A} \frac{A \sigma_{R,1} N F_b^{cm}}{V} \right] \\
&= \frac{\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}}{\sigma_{s,1 \rightarrow 2}} \psi_1^\dagger + \frac{A \sigma_{R,1} F_b^{cm}}{\sigma_{s,1 \rightarrow 2} N_A}.
\end{aligned} \tag{5.23}$$

We use the orthogonality condition, Eq. (3.77), as the additional equation to solve for the adjoint fluxes:

$$\begin{aligned}
0 &= \left\langle \psi^\dagger G \psi \right\rangle_{E,\mathcal{D},\Omega} \\
&= \left\langle \psi_{1,T}^\dagger \frac{\chi_1}{4\pi} N_T \nu 4\pi [\sigma_{f,1} \psi_{1,T} + \sigma_{f,2} \psi_{2,T}] + \psi_{2,T}^\dagger \frac{\chi_2}{4\pi} N_T \nu 4\pi [\sigma_{f,1} \psi_{1,T} + \sigma_{f,2} \psi_{2,T}] \right\rangle_{E,\mathcal{D},\Omega} \\
\rightarrow \psi_2^\dagger &= -\frac{\chi_1}{\chi_2} \psi_1^\dagger.
\end{aligned} \tag{5.24}$$

Combining Eqs. (5.23) and (5.24) yields:

$$\begin{aligned}
\psi_1^\dagger &= -\frac{\chi_2 A \sigma_{R,1} F_b^{\text{cm}}}{N_A (\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2})} \\
\psi_2^\dagger &= \frac{\chi_1 A \sigma_{R,1} F_b^{\text{cm}}}{N_A (\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2})}
\end{aligned}$$

Here we used Eq. (3.77) to directly solve for the adjoint fluxes; in practice, however, we will use a method of successive approximation (MSA) to creep up on the solution to the adjoint transport equation while maintaining the orthogonality to the forward fission source. To sweep out any fundamental mode contamination, we will use

$$\psi^\dagger = \psi_\ell^\dagger - \frac{\left\langle \psi_\ell^\dagger, G \psi \right\rangle_{E,\mathcal{D},\Omega}}{\left\langle \psi_H^\dagger, G \psi \right\rangle_{E,\mathcal{D},\Omega}} \psi_H^\dagger$$

where ψ_ℓ^\dagger is the ℓ^{th} iteration in a the MSA solution procedure, and ψ_H^\dagger is the solution to the homogeneous adjoint k-eigenvalue problem,

$$H^\dagger \psi_H^\dagger - \lambda G^\dagger \psi_H^\dagger = 0.$$

Here we quickly write the form of the solution ψ_H^\dagger . The homogeneous eigenvalue

problem equations are

$$\begin{aligned}\sigma_{t,1}N\psi_{H,1}^\dagger - N(\psi_{H,1}^\dagger\sigma_{s,1\rightarrow 1} + \psi_{H,2}^\dagger\sigma_{s,1\rightarrow 2}) - \lambda^\dagger\nu\sigma_{f,1}N(\psi_{H,1}^\dagger\chi_1 + \psi_{H,2}^\dagger\chi_2) &= 0 \\ \sigma_{t,2}N\psi_{H,2}^\dagger - N(\psi_{H,1}^\dagger\sigma_{s,2\rightarrow 1} + \psi_{H,2}^\dagger\sigma_{s,2\rightarrow 2}) - \lambda^\dagger\nu\sigma_{f,2}N(\psi_{H,1}^\dagger\chi_1 + \psi_{H,2}^\dagger\chi_2) &= 0.\end{aligned}$$

Here λ^\dagger is the fundamental adjoint mode eigenvalue, which we have shown before and will show below is equal to the fundamental forward mode eigenvalue. The two eigenfunctions are

$$\begin{aligned}\psi_{H,1}^\dagger &= \frac{\sigma_{s,1\rightarrow 2}}{\sigma_{t,1} - \sigma_{s,1\rightarrow 1}}\psi_{H,2}^\dagger \\ \psi_{H,1}^\dagger &= \frac{\sigma_{t,2} - \sigma_{s,2\rightarrow 2} - \lambda^\dagger\nu\sigma_{f,2}\chi_2}{\sigma_{s,2\rightarrow 1} + \lambda^\dagger\nu\sigma_{f,2}\chi_1}\psi_{H,2}^\dagger,\end{aligned}$$

the first of which does not change sign and is therefore the fundamental mode. Substituting this expression into the $g = 2$ homogeneous adjoint transport equation, we derive an expression for the fundamental adjoint eigenvalue, λ_0^\dagger :

$$\begin{aligned}\left[-\sigma_{s,2\rightarrow 1} - \lambda_0^\dagger\nu\sigma_{f,2}\chi_1\right]\psi_{H,1}^\dagger + \left[\sigma_{t,2} - \sigma_{s,2\rightarrow 2} - \lambda_0^\dagger\nu\sigma_{f,2}\chi_2\right]\psi_{H,2}^\dagger &= 0 \\ \left[-\sigma_{s,2\rightarrow 1} - \lambda_0^\dagger\nu\sigma_{f,2}\chi_1\right]\frac{\sigma_{s,1\rightarrow 2}}{\sigma_{t,1} - \sigma_{s,1\rightarrow 1}}\psi_{H,2}^\dagger + \left[\sigma_{t,2} - \sigma_{s,2\rightarrow 2} - \lambda_0^\dagger\nu\sigma_{f,2}\chi_2\right]\psi_{H,2}^\dagger &= 0 \\ \nu\sigma_{f,2}\left[\frac{-\chi_1\sigma_{s,1\rightarrow 2}}{\sigma_{t,1} - \sigma_{s,1\rightarrow 1}} - \chi_2\right]\lambda_0^\dagger &= -(\sigma_{t,2} - \sigma_{s,2\rightarrow 2}) + \frac{\sigma_{s,1\rightarrow 2}\sigma_{s,2\rightarrow 1}}{\sigma_{t,1} - \sigma_{s,1\rightarrow 1}} \\ \rightarrow \lambda_0^\dagger &= \frac{(\sigma_{t,1} - \sigma_{s,1\rightarrow 1})(\sigma_{t,2} - \sigma_{s,2\rightarrow 2}) - \sigma_{s,1\rightarrow 2}\sigma_{s,2\rightarrow 1}}{\nu\sigma_{f,2}\left[\chi_1\sigma_{s,1\rightarrow 2} + \chi_2(\sigma_{t,1} - \sigma_{s,1\rightarrow 1})\right]} \\ &= \lambda_0.\end{aligned}$$

The fundamental adjoint mode eigenvalue is indeed equal to the fundamental forward mode eigenvalue. The procedure in practice would be to solve the homogeneous

eigenvalue problem at each time stage, then perform the MSA solve, periodically sweeping out ψ_H^\dagger .

We have the terms necessary to integrate the sensitivity equation over the single time step in order to compute the derivatives given in Eqs. (5.12) through (5.19). The explicit Euler rule to integrate Eq. (3.85) backwards in time is

$$\int_{t_0}^{t_f} f(t) dt = Tf(T)$$

For $\frac{dQ}{d\sigma_{t,g}}$, the form of Eq. (3.85) is

$$\int_{t_0}^{t_f} \left\{ - \left\langle \psi^\dagger \frac{\partial H \psi}{\partial \sigma_{t,g}} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle N^\dagger \frac{\partial b N}{\partial \sigma_{t,g}} \right\rangle_{E,\mathcal{D},\Omega} \right\} dt$$

We evaluate these terms at $t = T$; the $g = 1$ terms, in detail, are

$$\begin{aligned}
\left\langle \psi^\dagger \frac{\partial H\psi}{\partial \sigma_{t,1}}(T) \right\rangle_{E,\mathcal{D},\Omega} &= 4\pi \psi_{1,T}^\dagger N_T \psi_{1,T} \\
&= 4\pi V \left[\frac{A_T \chi_2 \sigma_{R,1} F_b^{\text{cm}}}{N_A (\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2})} \right] N_T \left[\frac{\chi_1 \sigma_{R,2} + \sigma_{s,2 \rightarrow 1}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \right] \\
&= -4\pi V N_T \left[\frac{\mathcal{P}_0}{4\pi N_T E_f \sigma_{f,2} \psi_{2,T}} \right] \left[\frac{\chi_2 \sigma_{R,1} F_b^{\text{cm}}}{N_A (\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2})} \right] \\
&\quad \left[\frac{\chi_1 \sigma_{R,2} + \sigma_{s,2 \rightarrow 1}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \right] \\
&= -\frac{V \mathcal{P}_0 \chi_2 \sigma_{R,1} F_b^{\text{cm}} (\chi_2 \sigma_{R,2} + \sigma_{s,2 \rightarrow 1})}{E_f \sigma_{f,2} N_A [\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}]^2} \\
\left\langle N^\dagger \frac{\partial bN}{\partial \sigma_{t,1}}(T) \right\rangle_{E,\mathcal{D},\Omega} &= \left\langle -\frac{V}{N_A} \frac{A_T N_T \psi_{1,T} F_b^{\text{cm}}}{V} \right\rangle_{E,\mathcal{D},\Omega} \\
&= -\frac{4\pi V A_T N_T \psi_{1,T} F_b^{\text{cm}}}{N_A} \\
&= -\left[\frac{4\pi V F_b^{\text{cm}}}{N_A} \right] \left[\frac{\mathcal{P}_0}{4\pi N_T E_f \sigma_{f,2} \psi_{2,T}} \right] \left[\frac{\chi_1 \sigma_{R,2} + \sigma_{s,2 \rightarrow 1}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \right] N_T \psi_{2,T} \\
&= -\frac{V \mathcal{P}_0 \chi_2 \sigma_{R,1} F_b^{\text{cm}} (\chi_1 \sigma_{R,2} + \sigma_{s,2 \rightarrow 1})}{N_A E_f \sigma_{f,2} (\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2})} \\
&= -\frac{V \mathcal{P}_0 F_b^{\text{cm}} (\chi_1 \sigma_{R,2} + \sigma_{s,2 \rightarrow 1}) (\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2})}{N_A E_f \sigma_{f,2} [\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}]^2}
\end{aligned}$$

Using our integration rule, these terms combine to form the analytic derivative given in Eq. (5.12):

$$\frac{dQ}{d\sigma_{t,1}} = -\frac{TV \mathcal{P}_0 F_b^{\text{cm}}}{E_f N_A \sigma_{f,2}} \left[\frac{\chi_1 \sigma_{R,2} \sigma_{s,1 \rightarrow 2} + \sigma_{s,1 \rightarrow 2} \sigma_{s,2 \rightarrow 1}}{[\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}]^2} \right]$$

After similar substitutions, the $g = 2$ terms are

$$\begin{aligned}
\left\langle \psi^\dagger \frac{\partial H\psi}{\partial \sigma_{t,2}}(T) \right\rangle_{E,\mathcal{D},\Omega} &= \frac{V F_b^{\text{cm}} \mathcal{P}_0 (\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2})}{N_A E_f \sigma_{f,2} (\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2})} \\
\left\langle N^\dagger \frac{\partial bN}{\partial \sigma_{t,2}}(T) \right\rangle_{E,\mathcal{D},\Omega} &= -\frac{TV \mathcal{P}_0 F_b^{\text{cm}} (\sigma_{R,1} + \sigma_{s,1 \rightarrow 2})}{N_A E_f \sigma_{f,2} (\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2})}
\end{aligned}$$

These terms combine to form the expected result given in Eq. (5.13):

$$\frac{dQ}{d\sigma_{t,2}} = -\frac{TV\mathcal{P}_0F_b^{\text{cm}}(\sigma_{R,1} + \sigma_{s,1\rightarrow 2})}{E_f N_A \sigma_{f,2} (\chi_2 \sigma_{R,1} + \sigma_{s,1\rightarrow 2})}$$

Now for the scattering cross sections. The form of the sensitivity equation is as follows:

$$\begin{aligned} \frac{dQ}{d\sigma_{s,g' \rightarrow g}} &= \int_{t_0}^{t_f} -\left\langle \psi^\dagger \frac{\partial H\psi}{\partial \sigma_{s,g' \rightarrow g}} \right\rangle_{E,\mathcal{D},\Omega} dt \\ &= \int_{t_0}^{t_f} -4\pi V \psi_g^\dagger (-N\psi_{g'}) dt \\ &= \int_{t_0}^{t_f} 4\pi V N \psi_g^\dagger \psi_{g'} dt \end{aligned}$$

For example, consider the case of $g' = g = 1$ using our explicit Euler integration rule:

$$\begin{aligned} \frac{dQ}{d\sigma_{s,1\rightarrow 1}} &= \int_{t_0}^{t_f} 4\pi V N \psi_1^\dagger \psi_1 dt \\ &\equiv 4\pi TV N_T \psi_{1,T}^\dagger \psi_{1,T} \\ &= 4\pi TV \left[-\frac{A_T \chi_2 \sigma_{R,1} F_b^{\text{cm}}}{N_A (\chi_2 \sigma_{R,1} + \sigma_{s,1\rightarrow 2})} \right] N_T \left[\frac{\chi_1 \sigma_{R,2} + \sigma_{s,2\rightarrow 1}}{\chi_2 \sigma_{R,1} + \sigma_{s,1\rightarrow 2}} \right] \psi_{2,T} \\ &= -4\pi TV N_T \psi_{2,T} \left[\frac{\mathcal{P}_0}{4\pi N_T E_f \sigma_{f,2} \psi_{2,T}} \right] \left[\frac{\chi_2 \sigma_{R,1} F_b^{\text{cm}} (\chi_1 \sigma_{R,2} + \sigma_{s,2\rightarrow 1})}{N_A (\chi_2 \sigma_{R,1} + \sigma_{s,1\rightarrow 2})^2} \right] \\ &= -\frac{\mathcal{P}_0 F_b^{\text{cm}} TV}{E_f \sigma_{f,2} N_A} \left[\frac{\chi_2 \sigma_{R,1} (\chi_1 \sigma_{R,2} + \sigma_{s,2\rightarrow 1})}{[\chi_2 \sigma_{R,1} + \sigma_{s,1\rightarrow 2}]^2} \right] \end{aligned}$$

which is equivalent to Eq. (5.14). Similar manipulations and integrations lead to Eqs. (5.15) through (5.17). Next we consider the form of Eq. (3.85) for computing $\frac{dQ}{d\nu}$. This parameter appears only in the fission operator; the sensitivity equation is therefore

$$\frac{dQ}{d\nu} = \int_{t_0}^{t_f} \lambda \left\langle \psi^\dagger \frac{\partial G\psi}{\partial \nu} \right\rangle_{E,\mathcal{D},\Omega} dt.$$

The fission source for group g is $G\psi = \chi_g N \nu \sigma_{f,2} \psi_2$; therefore

$$\begin{aligned}\left\langle \psi^\dagger \frac{\partial G\psi}{\partial \nu} \right\rangle_{E,\mathcal{D},\Omega} &= 4\pi V \left[\psi_1^\dagger \chi_1 N \sigma_{f,2} \psi_2 + \psi_2^\dagger \chi_2 N \sigma_{f,2} \psi_2 \right] \\ &= 4\pi V N \sigma_{f,2} \psi_2 \left[\psi_1^\dagger \chi_1 + \psi_2^\dagger \chi_2 \right] \\ &= 0,\end{aligned}$$

which gives our expected result, Eq. (5.18). Now consider the form of Eq. (3.85) for

$$\frac{dQ}{d\sigma_{f,2}}:$$

$$\frac{dQ}{d\sigma_{f,2}} = \int_{t_0}^{t_f} \left\{ \lambda \left\langle \psi^\dagger \frac{\partial G\psi}{\partial \sigma_{f,2}} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A^\dagger \frac{A}{V} \frac{\partial \Sigma_E \psi}{\partial \sigma_{f,2}} \right\rangle_{E,\mathcal{D},\Omega} \right\} dt$$

Using $\psi_2^\dagger = -\frac{\chi_1}{\chi_2} \psi_1^\dagger$ we have

$$\begin{aligned}\lambda \left\langle \psi^\dagger \frac{\partial G\psi}{\partial \sigma_{f,2}} \right\rangle_{E,\mathcal{D},\Omega} &= 4\pi V \lambda \left[\psi_{1,T}^\dagger \chi_1 N_T \nu \psi_{2,T} + \psi_{2,T}^\dagger \chi_2 N_T \nu \psi_{2,T} \right] \\ &= 4\pi V \lambda \left[\psi_{1,T}^\dagger \chi_1 N_T \nu \psi_{2,T} - \psi_{1,T}^\dagger \frac{\chi_1}{\chi_2} \chi_2 N_T \nu \psi_{2,T} \right] \\ &= 0.\end{aligned}$$

For the second term, we find

$$\begin{aligned}\left\langle A_T^\dagger \frac{A_T}{V} \frac{\partial \Sigma_E \psi(T)}{\partial \sigma_{f,2}} \right\rangle_{E,\mathcal{D},\Omega} &= A_T^\dagger \frac{A_T}{V} \left\langle \frac{\partial \Sigma_{E,g} \psi}{\partial \sigma_{f,2}} \right\rangle_{E,\mathcal{D},\Omega} \\ &= A_T^\dagger \frac{A_T}{V} 4\pi V E_f \psi_{2,T} N_T \\ &= -\frac{4\pi F_b^{\text{cm}} V \psi_{2,T} E_f}{N_A V E_f \sigma_{f,2}} \left[\frac{\sigma_{R,1} \sigma_{R,2} + \sigma_{R,1} \sigma_{s,2 \rightarrow 1} + \sigma_{R,2} \sigma_{s,1 \rightarrow 2}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \right] \\ &\quad \left[\frac{\mathcal{P}_0}{4\pi E_f N_T \sigma_{f,2} \psi_{2,T}} \right]\end{aligned}$$

After cancellation and integration, this reduces to Eq. (5.19):

$$\frac{dQ}{d\sigma_{f,2}} = \frac{TV\mathcal{P}_0F_b^{\text{cm}}}{E_f N_A \sigma_{f,2}^2} \left[\frac{\sigma_{R,1}\sigma_{R,2} + \sigma_{R,1}\sigma_{s,2 \rightarrow 1} + \sigma_{R,2}\sigma_{s,1 \rightarrow 2}}{\chi_2\sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \right]$$

To compute the sensitivity of our QOI with respect to the initial condition, we must integrate the adjoint equations from $t = T$ to $t = 0$ to compute $N^\dagger(0)$. First we solve the adjoint Bateman equation, which from Eq. (3.75) is

$$\frac{dN^\dagger}{dt} = \left\langle \psi^\dagger \frac{\partial H\psi}{\partial N} - \lambda\psi^\dagger \frac{\partial G\psi}{\partial N} - b^\dagger N^\dagger + A^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega}.$$

Our integration rule is

$$N_0^\dagger = N_T^\dagger - Tf(T)$$

where $f(T)$ is the right-hand side of the adjoint Bateman equation. Let's look term-

by-term. First (all terms evaluated at $t = T$),

$$\begin{aligned}
\left\langle \psi^\dagger \frac{\partial H\psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} &= 4\pi V \left[\psi_1^\dagger (\sigma_{t,1}\psi_1 - \sigma_{s,1 \rightarrow 1}\psi_1 - \sigma_{s,2 \rightarrow 1}\psi_2) \right. \\
&\quad \left. + \psi_2^\dagger (\sigma_{t,2}\psi_2 - \sigma_{s,1 \rightarrow 2}\psi_1 - \sigma_{s,2 \rightarrow 2}\psi_2) \right] \\
&= 4\pi V \psi_1^\dagger \left[\sigma_{t,1}\psi_1 - \sigma_{s,1 \rightarrow 1}\psi_2 - \sigma_{s,2 \rightarrow 1}\psi_2 \right. \\
&\quad \left. - \frac{\chi_1}{\chi_2} (\sigma_{t,2}\psi_2 - \sigma_{s,1 \rightarrow 2}\psi_1 - \sigma_{s,2 \rightarrow 2}\psi_2) \right] \\
&= 4\pi V \psi_1^\dagger \left[\frac{\chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}) + \chi_1\sigma_{s,1 \rightarrow 2}}{\chi_2} \psi_1 \right. \\
&\quad \left. + \frac{-\chi_1(\sigma_{t,2} - \sigma_{s,2 \rightarrow 2}) - \chi_2\sigma_{s,2 \rightarrow 1}}{\chi_2} \psi_2 \right] \\
&= \frac{4\pi V \psi_1^\dagger \psi_2}{\chi_2} \left[-\chi_1(\sigma_{t,2} - \sigma_{s,2 \rightarrow 2}) - \chi_2\sigma_{s,2 \rightarrow 1} \right. \\
&\quad \left. + (\chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}) + \chi_1\sigma_{s,1 \rightarrow 2}) \frac{\chi_1(\sigma_{t,2} - \sigma_{s,2 \rightarrow 2}) + \chi_2\sigma_{s,2 \rightarrow 1}}{\chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}) + \chi_1\sigma_{s,1 \rightarrow 2}} \right] \\
&= 0.
\end{aligned}$$

The fission term also evaluates to zero:

$$\begin{aligned}
\left\langle \lambda \psi^\dagger \frac{\partial G\psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} &= 4\pi V \lambda \left[\psi_1^\dagger \chi_1 \nu \sigma_{f,2} \psi_2 + \psi_2^\dagger \chi_2 \nu \sigma_{f,2} \psi_2 \right] \\
&= 4\pi V \lambda \nu \sigma_{f,2} \psi_2 \left[\psi_1^\dagger \chi_1 + \psi_2^\dagger \chi_2 \right] \\
&= 0.
\end{aligned}$$

This leaves (again all terms evaluated at $t = T$)

$$\begin{aligned}
\frac{dN^\dagger}{dt}(T) &= \left\langle \left[-b^\dagger N^\dagger + A^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial N} \right] \right\rangle_{E,\mathcal{D},\Omega} \\
&= \left\langle \frac{A \sigma_{t,g} \psi_g F_b^{\text{cm}} N^\dagger}{V_R} + A^\dagger \frac{A}{V_R} E_f \sigma_{f,g} \psi_g \right\rangle_{E,\mathcal{D},\Omega} \\
&= 4\pi V A \left[\frac{F_b^{\text{cm}}}{N_A} [\psi_1 \sigma_{R,1} + \psi_2 \sigma_{R,2}] + \frac{A^\dagger}{V} E_f \sigma_{f,2} \psi_2 \right] \\
&= 4\pi V A \left[\frac{F_b^{\text{cm}}}{N_A} \left[\frac{\chi_1 \sigma_{R,2} + \sigma_{s,2 \rightarrow 1}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \sigma_{R,1} + \sigma_{R,2} \right] \psi_2 \right. \\
&\quad \left. - \frac{F_b^{\text{cm}} V}{N_A E_f \sigma_{f,2}} \left[\frac{\sigma_{R,1} \sigma_{R,2} + \sigma_{R,1} \sigma_{s,2 \rightarrow 1} + \sigma_{R,2} \sigma_{s,1 \rightarrow 2}}{\chi_2 \sigma_{R,1} + \sigma_{s,1 \rightarrow 2}} \right] \frac{E_f \sigma_{f,2} \psi_2}{V} \right] \\
&= 0.
\end{aligned}$$

Thus, the adjoint densities are constant in time: $N_0^\dagger = N_T^\dagger = \frac{V}{N_A}$. It follows, then, that $A_0^\dagger = A_T^\dagger$ and $\psi_{g,0}^\dagger = \psi_{g,T}^\dagger$.

The final step is to use Eq. (3.85) to compute the sensitivity of the QOI with respect to the initial condition. The parameter N_0 only appears in the term $\left[N^\dagger \frac{dN}{dp} \right]_{t=t_0}$.

At $t = t_0$, $\frac{dN}{dN_0} = 1$, thus

$$\frac{dQ}{dN_0} = N_0^\dagger = \frac{V}{N_A},$$

which agrees with the analytic result.

This problem was run in PDT to verify that the numerical results agree with the predictions made above. The value of the material and geometry parameters are given in Table 5.6.

Table 5.6: Parameters for infinite medium, 2 group k-eigenvalue adjoint verification problem

Parameter	Value
# Cells	1
Cell Volume, V	$(2.0 \times 3.0 \times 2.0) \cdot 10^7 = 1.2\text{e}+22 \text{ cm}^3$
Spatial Discretization	PWLD (8 Elements)
Element Volume, V_e	1.5e+21 cm ³
Simulation Time, T	5.0e+06s \approx 8.25 weeks
Runge-Kutta Scheme	Explicit Euler
Initial Density, N_0	5.0 [$\frac{\text{atom}}{\text{b}-\text{cm}}$]
$\sigma_{s,1 \rightarrow 1}, \sigma_{s,1 \rightarrow 2}, \sigma_{s,2 \rightarrow 1}, \sigma_{s,2 \rightarrow 2}$	0.6, 0.2, 0.05, 1.4 [b]
$\sigma_{t,1}, \sigma_{t,2}, \sigma_{f,2}$	2.0, 5.3, 3.38 [b]
χ_1, χ_2, ν	0.85, 0.15, 2.23
E_f, \mathcal{P}_0	206.0 MeV, 60.0 W/cm ³
Fixed-source solver tolerance	1.0e-07
Eigenvalue, eigenvector tolerance	1.0e-07, 1.0e-06

Table 5.7 gives a list of intermediate quantities for this problem, their expected numerical value, the numerical value computed in PDT, and the relative difference between the two results.

Table 5.7: Comparison of expected and calculated numerical results for the infinite medium, 2 group k-eigenvalue adjoint depletion verification problem (ϵ =machine precision).

Description	Exected Value	PDT Value	Relative Difference
$\frac{\psi_{1,0}}{\psi_{2,0}}$	8.74342105e+00	8.74342091e+00	1.61e-08
$A(0)$	6.69384179e+11	6.69384153e+11	3.95e-08
$k_{\text{eff}}(0)$	5.25543486e-01	5.25543481e-01	9.36e-09
N_T	4.99996143e+00	4.99996143e+00	ϵ
$\frac{\psi_{1,T}}{\psi_{2,T}}$	8.74342105e+00	8.74342091e+00	1.65e-08
$A(T)$	6.69389343e+11	6.69389317e+11	3.94e-08
$k_{\text{eff}}(T)$	5.25543486e-01	5.25543482e-01	8.90e-09
$Q = \text{QOI}$	9.96315667e+22	9.96315667e+22	ϵ
N_T^\dagger	1.99264671e+22	1.99264671e+22	ϵ
A_T^\dagger	-2.56182192e+09	-2.56182191e+09	-1.56e-09
$S_{1,T}^\dagger$	-6.66923090e-12	-6.66923084e-12	8.72e-09
$S_{2,T}^\dagger$	5.83118939e-11	5.83118932e-11	1.09e-08
$\psi_{1,T}^\dagger$	-5.26522291e-13	-5.26522254e-13	-6.86e-08
$\psi_{2,T}^\dagger$	2.98362631e-12	2.98362615e-12	5.42e-08
N_0^\dagger	1.99264671e+22	1.99264671e+22	ϵ
A_0^\dagger	-2.56182192e+09	-2.56182191e+09	-1.45e-09
$\psi_{1,0}^\dagger$	-5.26522291e-13	-5.26522284e-13	-7.78e-06
$\psi_{2,0}^\dagger$	2.98362631e-12	2.98362626e-12	7.77e-06

Finally, Table 5.8 gives the expected and computed values of the parameter derivatives, and the relative difference between them.

Table 5.8: Comparison of expected and calculated numerical derivatives for the infinite medium, 2 group k-eigenvalue adjoint depletion verification problem(ϵ =machine precision).

Description	Exected Value	PDT Value	Relative Difference
$\frac{dQ}{d\sigma_{t,1}}$	-2.46595461E+17	-2.46593749E+17	-6.94e-06
$\frac{dQ}{d\sigma_{t,2}}$	-1.97424808E+17	-1.97425917E+17	-5.62E-06
$\frac{dQ}{d\sigma_{s,1 \rightarrow 1}}$	-2.21935915E+17	-2.21937625E+17	-7.71E-06
$\frac{dQ}{d\sigma_{s,1 \rightarrow 2}}$	1.25763685E+18	1.25764654E+18	7.71E-06
$\frac{dQ}{d\sigma_{s,2 \rightarrow 1}}$	-2.53831896E+16	-2.53833853E+16	-7.71E-06
$\frac{dQ}{d\sigma_{s,2 \rightarrow 2}}$	1.43838075E+17	1.43839183E+17	7.71E-06
$\frac{dQ}{d\nu}$	0.0	3.54422653e+01	ϵ
$\frac{dQ}{d\sigma_{f,2}}$	2.27380643E+17	2.27380643E+17	1.74E-09
$\frac{dQ}{dN_0}$	1.99264671E+22	1.99264671E+22	ϵ

This verification problem demonstrates that the code is producing a discrete solution that matches, to the order of the solver tolerances, by-hand discrete solutions for a k-eigenvalue, infinite medium, 2-group problem. A similar step-by-step walk through for a simple source-driven problem is described in Sec. C.1.

We now walk through a similar analysis for a flux dependent QOI: the total reaction rate in the *very large* medium. The flux dependent QOI requires one additional fixed-source solve, which we outline here. Using the form of Eq. (3.74), our QOI is

$$\begin{aligned}
 Q &= \left\langle R(t_f) \right\rangle_{E,\mathcal{D},\Omega} = \left\langle A\psi\sigma_t N \right\rangle_{E,\mathcal{D},\Omega} = 4\pi V A(T) N_T \left[\psi_1(T)\sigma_{t,1} + \psi_2(T)\sigma_{t,2} \right] \\
 &= \frac{V\mathcal{P}_0}{E_f\sigma_{f,2}} \left[\gamma\sigma_{t,1} + \sigma_{t,2} \right],
 \end{aligned}$$

where

$$\gamma = \frac{\psi_1}{\psi_2} = \frac{\chi_1(\sigma_{t,2} - \sigma_{s,2 \rightarrow 2}) + \chi_2 \sigma_{s,2 \rightarrow 1}}{\chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}) + \chi_1 \sigma_{s,1 \rightarrow 2}}.$$

As above, we will verify that the adjoint formalism produces the following subset of analytic derivatives of the discrete QOI:

$$\frac{dQ}{d\sigma_{t,1}} = \frac{V\mathcal{P}_0\gamma}{E_f\sigma_{f,2}} \left[1 - \frac{\chi_2\sigma_{t,1}}{\chi_1\sigma_{s,1 \rightarrow 2} + \chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})} \right] \quad (5.25)$$

$$\frac{dQ}{d\sigma_{s,1 \rightarrow 1}} = \frac{V\mathcal{P}_0\sigma_{t,1}\gamma\chi_2}{E_f\sigma_{f,2}[\chi_1\sigma_{s,1 \rightarrow 2} + \chi_2(\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})]} \quad (5.26)$$

$$\frac{dQ}{d\sigma_{f,2}} = -\frac{V\mathcal{P}_0}{E_f\sigma_{f,2}^2} [\gamma\sigma_{t,1} + \sigma_{t,2}]. \quad (5.27)$$

To figure the terminal adjoint density, we first evaluate \hat{A}^\dagger using Eq. (3.82):

$$\begin{aligned} \hat{A}^\dagger &= \frac{V}{\hat{\mathcal{P}}} \left\langle \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} = \frac{V}{\left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega}} \left\langle \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} \\ &= \frac{V}{4\pi V N_T \sigma_{f,2} E_f \psi_{2,T}} 4\pi V N_T [\sigma_{t,1} \gamma \psi_{2,t} + \sigma_{t,2} \psi_{2,T}] \\ &= \frac{V [\gamma \sigma_{t,1} + \sigma_{t,2}]}{E_f \sigma_{f,2}} \end{aligned}$$

Turning to $\hat{\psi}^\dagger$, we write the group 1 version of Eq. (3.83):

$$\sigma_{t,1} N_T \hat{\psi}_1^\dagger - \sigma_{s,1 \rightarrow 1} N_T \hat{\psi}_1^\dagger - \sigma_{s,1 \rightarrow 2} N_T \hat{\psi}_2^\dagger = \frac{\partial R}{\partial \psi} = A_T \sigma_{t,1} N_T. \quad (5.28)$$

Using Eq. (3.80), we again have $\hat{\psi}_1^\dagger = -\frac{\chi_2}{\chi_1} \hat{\psi}_2^\dagger$. Substitution of this expression into

Eq. (5.28), we find the following values for $\hat{\psi}^\dagger$:

$$\begin{aligned}\hat{\psi}_1^\dagger &= \frac{\chi_2 A_T \sigma_{t,1}}{\chi_1 \sigma_{s,1 \rightarrow 2} + \chi_2 (\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})} \\ \hat{\psi}_2^\dagger &= -\frac{\chi_1 A_T \sigma_{t,1}}{\chi_1 \sigma_{s,1 \rightarrow 2} + \chi_2 (\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})}.\end{aligned}$$

We now have the information necessary to evaluate Eq. (3.84) for the terminal adjoint densities. The expression is

$$N^\dagger(T) = \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \hat{\psi}^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} - \hat{A}^\dagger \frac{A(T)}{V} \frac{\partial \hat{\mathcal{P}}}{\partial N}.$$

The first term is

$$\frac{\partial R}{\partial N} = 4\pi V A_T [\gamma \sigma_{t,1} \psi_{2,T} + \sigma_{t,2} \psi_{2,T}] = \frac{V \mathcal{P}_0 [\gamma \sigma_{t,1} + \sigma_{t,2}]}{N E_f \sigma_{f,2}}.$$

The third term is

$$\begin{aligned}\hat{A}^\dagger \frac{A_T}{V} \frac{\partial}{\partial N} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} &= \frac{V [\gamma \sigma_{t,1} + \sigma_{t,2}]}{E_f \sigma_{f,2}} \frac{A_T}{V} \frac{\partial}{\partial N} [4\pi V N_T E_f \sigma_{f,2} \psi_{2,T}] \\ &= \frac{V \mathcal{P}_0 [\gamma \sigma_{t,1} + \sigma_{t,2}]}{N E_f \sigma_{f,2}}.\end{aligned}$$

Thus

$$\begin{aligned}
N^\dagger(T) &= - \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \\
&= -4\pi V \left[\frac{\chi_1 A_T \sigma_{t,1}}{\chi_2 \sigma_{s,1 \rightarrow 2} + \chi_2 (\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})} \right. \\
&\quad \left((\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}) \psi_{1,T} - (\sigma_{s,2 \rightarrow 1} + \lambda \chi_1 \nu \sigma_{f,2}) \psi_{2,T} \right) \\
&\quad - \frac{\chi_1 A_T \sigma_{t,1}}{\chi_1 \sigma_{s,1 \rightarrow 2} + \chi_2 (\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})} \\
&\quad \left. \left(-\sigma_{s,1 \rightarrow 2} \psi_{1,T} + (\sigma_{t,2} - \sigma_{s,2 \rightarrow 2} - \lambda \chi_1 \nu \sigma_{f,2}) \psi_{2,T} \right) \right] \\
&= - \frac{4\pi V A_T \sigma_{t,1}}{\chi_1 \sigma_{s,1 \rightarrow 2} + \chi_2 (\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})} \left[\chi_2 (\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}) \gamma \psi_{2,T} - \chi_2 \sigma_{s,2 \rightarrow 1} \psi_{2,T} \right. \\
&\quad \left. + \chi_1 \sigma_{s,1 \rightarrow 2} \gamma \psi_{2,T} - (\sigma_{t,2} - \chi_1 \sigma_{s,2 \rightarrow 2}) \psi_{2,T} \right] \\
&= - \frac{V \mathcal{P}_0 \sigma_{t,1}}{N_T E_f \sigma_{f,2} [\chi_1 \sigma_{s,1 \rightarrow 2} + \chi_2 (\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})]} \\
&\quad \left(\gamma [\chi_2 (\sigma_{t,1} - \sigma_{s,1 \rightarrow 1}) + \chi_1 \sigma_{s,1 \rightarrow 2}] - [\chi_2 \sigma_{s,2 \rightarrow 1} + \chi_1 (\sigma_{t,2} - \sigma_{s,2 \rightarrow 2})] \right) \\
&= 0
\end{aligned}$$

The terminal adjoint densities are zero. It follows from Eqs. (3.78) and (3.76) that $A^\dagger(T) = 0$ and $\psi^\dagger(T) = 0$, respectively. Thus, the only non-zero terms in the integration of the sensitivity equation, Eq. (3.85), are

$$\begin{aligned}
\frac{dQ}{dp} &= \left[N^\dagger \frac{dN}{dt} \right]_{t=t_0} + \left[\left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \right]_{t=t_f} \\
&\quad - \left[\left\langle \hat{\psi}^\dagger, \frac{\partial}{\partial p} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} + \hat{A}^\dagger \frac{\partial}{\partial p} \left[\frac{A}{V} \hat{\mathcal{P}} - \mathcal{P} \right] \right]_{t=t_f}. \quad (5.29)
\end{aligned}$$

The form of Eq. (5.29) for $\sigma_{t,g}$ is

$$\begin{aligned}\frac{dQ}{d\sigma_{t,g}} &= \left\langle \frac{\partial R}{\partial \sigma_{t,g}} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \hat{\psi}^\dagger, \frac{\partial}{\partial \sigma_{t,g}} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \\ &= 4\pi V A_T N_T \psi_{g,T} - 4\pi V \hat{\psi}_g^\dagger \psi_{g,T} N_T\end{aligned}$$

For $\sigma_{t,1}$, we find this leads to an expression identical to Eq. (5.25):

$$\begin{aligned}\frac{dQ}{d\sigma_{t,1}} &= 4\pi V A_T N_T \psi_{1,T} - 4\pi V \hat{\psi}_1^\dagger \psi_{1,T} N_T \\ &= \frac{V \mathcal{P}_0 \gamma}{E_f \sigma_{f,2}} - 4\pi V \frac{\chi_2 A_T \sigma_{t,1} \gamma \psi_{2,T} N_T}{\chi_1 \sigma_{s,1 \rightarrow 2} + \chi_2 (\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})} \\ &= \frac{V \mathcal{P}_0 \gamma}{E_f \sigma_{f,2}} - \frac{V \chi_2 \sigma_{t,1} \mathcal{P}_0 \gamma}{E_f \sigma_{f,2} [\chi_1 \sigma_{s,1 \rightarrow 2} + \chi_2 (\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})]} \\ &= \frac{V \mathcal{P}_0 \gamma}{E_f \sigma_{f,2}} \left[1 - \frac{\chi_2 \sigma_{t,1}}{\chi_1 \sigma_{s,1 \rightarrow 2} + \chi_2 (\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})} \right]\end{aligned}$$

For $\sigma_{s,g \rightarrow g'}$, Eq. (5.29) reduces to

$$\frac{dQ}{d\sigma_{s,g \rightarrow g'}} = - \left\langle \hat{\psi}_1^\dagger, \frac{\partial}{\partial \sigma_{s,g \rightarrow g'}} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega}$$

For $\sigma_{s,1 \rightarrow 1}$, we find an expression equal to the expected result, Eq. (5.26)

$$\begin{aligned}\frac{dQ}{d\sigma_{s,1 \rightarrow 1}} &= -4\pi V \left[\hat{\psi}_1^\dagger (-N_T \psi_{1,T}) \right] \\ &= 4\pi V \frac{\chi_2 A_T \sigma_{t,1} \gamma \psi_{2,T} N_T}{\chi_1 \sigma_{s,1 \rightarrow 2} + \chi_2 (\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})} \\ &= \frac{V \mathcal{P}_0 \sigma_{t,1} \gamma \chi_2}{E_f \sigma_{f,2} [\chi_1 \sigma_{s,1 \rightarrow 2} + \chi_2 (\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})]}\end{aligned}$$

The final example is for $\frac{dQ}{d\sigma_{f,2}}$. The non-zero terms in Eq. (5.29) are

$$\frac{dQ}{d\sigma_{f,2}} = \left\langle \hat{\psi}^\dagger, \frac{\partial}{\partial \sigma_{f,2}} [\lambda G \psi] \right\rangle_{E,\mathcal{D},\Omega} - \hat{A}^\dagger \frac{\partial}{\partial \sigma_{f,2}} \left[\frac{A}{V} \hat{\mathcal{P}} - \mathcal{P}_0 \right]$$

For the first term we find

$$\begin{aligned} \left\langle \hat{\psi}^\dagger, \frac{\partial}{\partial \sigma_{f,2}} [\lambda G \psi] \right\rangle_{E,\mathcal{D},\Omega} &= 4\pi V \left[\frac{\chi_2 A_T \sigma_{t,1}}{\chi_1 \sigma_{s,1 \rightarrow 2} + \chi_2 (\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})} \chi_1 \lambda \nu N_T \psi_{2,T} \right. \\ &\quad \left. - \frac{\chi_1 A_T \sigma_{t,1}}{\chi_1 \sigma_{s,1 \rightarrow 2} + \chi_2 (\sigma_{t,1} - \sigma_{s,1 \rightarrow 1})} \chi_2 \lambda \nu N_T \psi_{2,T} \right] \\ &= 0. \end{aligned}$$

For the second term we find

$$\begin{aligned} \hat{A}^\dagger \frac{\partial}{\partial \sigma_{f,2}} \left[\frac{A}{V} \hat{\mathcal{P}} - \mathcal{P}_0 \right] &= \frac{V [\gamma \sigma_{t,1} + \sigma_{t,2}]}{E_f \sigma_{f,2}} \frac{A}{V} 4\pi V N_T \psi_{2,T} E_f \\ &= \frac{V \mathcal{P}_0 [\gamma \sigma_{t,1} + \sigma_{t,2}]}{E_f \sigma_{f,2}^2} \end{aligned}$$

Applying the negative sign, we get a derivative that matches the expected result given by Eq. (5.27). The problem described by Table 5.6 was run again with the reaction rate QOI. Again, we obtained numerical results that were within solver tolerance of the expected results.

5.3.2 Convergence rates of the sensitivity and error estimates

Here we use an analytic solution to a slightly-modified forward problem to show convergence rates of the sensitivity and error estimates made by PDT. The modification is the addition of decay dynamics to the Bateman equation. The version of

Eqs. (5.3) – (5.7) we use here are

$$\begin{aligned}
\frac{dN}{dt} &= -4\pi ANF_b^{\text{cm}} [\sigma_{R,1}\psi_1 + \sigma_{R,2}\psi_2] - \lambda N & (5.30) \\
\sigma_{t,1}N\psi_1 - N(\psi_1\sigma_{s,1\rightarrow 1} + \psi_2\sigma_{s,2\rightarrow 1}) - \lambda_0\chi_1\nu N\sigma_{f,2}\psi_2 &= 0 \\
\sigma_{t,2}N\psi_2 - N(\psi_1\sigma_{s,1\rightarrow 2} + \psi_2\sigma_{s,2\rightarrow 2}) - \lambda_0\chi_2\nu N\sigma_{f,2}\psi_2 &= 0 \\
\mathcal{P}_0 - A4\pi NE_f\sigma_{f,2}\psi_2 &= 0 \\
N(0) = N_0, \quad t \in [0, t_f], &
\end{aligned}$$

where λ is the decay constant of the component. We find the same fundamental eigenvector as in Eq. (5.8):

$$\frac{\psi_1}{\psi_2} = \frac{\lambda_0\chi_1\nu\sigma_{f,2} + \sigma_{s,2\rightarrow 1}}{\sigma_{t,1} - \sigma_{s,1\rightarrow 1}}$$

with eigenvalue

$$\lambda_0 = \frac{(\sigma_{t,1} - \sigma_{s,1\rightarrow 1})(\sigma_{t,2} - \sigma_{s,2\rightarrow 2}) - \sigma_{s,1\rightarrow 2}\sigma_{s,2\rightarrow 1}}{\nu\sigma_{f,2}[\chi_1\sigma_{s,1\rightarrow 2} + \chi_2(\sigma_{t,1} - \sigma_{s,1\rightarrow 1})]}.$$

After substitution of these expressions and Eq. (5.10) back into Eq. (5.30), we find the following simple ODE governing the component density:

$$\begin{aligned}
\frac{dN}{dt} &= -\frac{\mathcal{P}_0 F_b^{\text{cm}}}{E_f \sigma_{f,2}} \left[\sigma_{R,1} \frac{\lambda_0 \chi_1 \nu \sigma_{f,2} + \sigma_{s,2\rightarrow 1}}{\sigma_{t,1} - \sigma_{s,1\rightarrow 1}} + \sigma_{R,2} \right] - \lambda N \\
&\equiv -\kappa - \lambda N. & (5.31)
\end{aligned}$$

This ODE has analytic solution

$$\begin{aligned} N(t) &= \left(N_0 + \frac{\kappa}{\lambda} \right) \exp(-\lambda t) - \frac{\kappa}{\lambda} \\ &= N_0 \exp(-\lambda t) - \frac{\kappa}{\lambda} [1 - \exp(-\lambda t)], \end{aligned}$$

yielding an analytic expressions for our nuclide inventory QOI, which we will use for the remainder of this analysis:

$$Q(t_f) = \frac{V}{N_A} \left(N_0 + \frac{\kappa}{\lambda} \right) \exp(-\lambda t_f) - \frac{\kappa V}{\lambda N_A} \quad (5.32)$$

$$= \frac{VN_0}{N_A} \exp(-\lambda t_f) - \frac{\kappa V}{\lambda N_A} [1 - \exp(-\lambda t_f)] \quad (5.33)$$

Now consider solving Eq. (5.31) over a series of time steps of width T using the explicit Euler rule. The scheme would proceed as follows:

$$\begin{aligned} N_1 &= N_0 - T(\kappa + \lambda N_0) \\ &= (1 - \lambda T)N_0 - \kappa T \\ N_2 &= N_1 - T(\kappa + \lambda N_1) \\ &= (1 - \lambda T)N_1 - \kappa T \\ &= (1 - \lambda T)^2 N_0 - \kappa T(1 - \lambda T) - \kappa T \\ N_3 &= N_2 - T(\kappa + \lambda N_2) \\ &= (1 - \lambda T)^3 N_0 - \kappa T(1 - \lambda T)^2 - \kappa T(1 - \lambda T) - \kappa T \\ &\dots \\ N_j &= (1 - \lambda T)^j N_0 - \kappa T \sum_{i=0}^{j-1} (1 - \lambda T)^i, \end{aligned}$$

where $N_j = N(t = jT)$. The numerical QOI is

$$Q_j = \frac{V}{N_A} \left[(1 - \lambda T)^j N_0 - \kappa T \sum_{i=0}^{j-1} (1 - \lambda T)^i \right]. \quad (5.34)$$

To confirm the order of accuracy of the scheme, we expand Eq. (5.32) about small t_f and find

$$Q(t_f) = \frac{V}{N_A} \left[N_0 - t_f (N_0 \lambda + \kappa) + \frac{1}{2} t_f^2 (N_0 \lambda^2 + \kappa \lambda) + \mathcal{O}(t_f^3) \right]. \quad (5.35)$$

Similarly we expand Eq. (5.34) about small T and find

$$\begin{aligned} Q_j(T) &= \frac{V}{N_A} \left[N_0 - jT (N_0 \lambda + \kappa) \right. \\ &\quad \left. + \frac{1}{2} \left(N_0 \lambda^2 j(j-1)T^2 + \kappa \lambda T^2 \sum_{i=0}^{j-1} i \right) + \mathcal{O}(T^3) \right]. \end{aligned} \quad (5.36)$$

By inspection of Eqs. (5.35) and (5.36), we see that the local truncation error is proportional to T^2 , a familiar result for the explicit Euler scheme. The global truncation error loses an order of accuracy, however, because of the $\mathcal{O}\left(\frac{1}{T}\right)$ number of time steps required to solve the problem. After a similar analysis for the modified Euler Runge-Kutta scheme, we find that the error in the discrete QOI approximation is proportional to T^3 , or that the modified Euler scheme is globally second order accurate, as expected.

This analysis, in combination with the preceding analysis that produced the discrete derivative expressions (5.12) – (5.20), lends the conclusion that the Runge-Kutta schemes should produce QOI derivative estimates that converge globally at the rate consistent with the scheme. From section 2.5.2, we know this convergence rate will be suboptimal if (a) the adjoint of the forward Runge-Kutta scheme is not used to

integrate the adjoint equations, or (b) if the scheme is not implemented correctly.

To test these convergence rates, the problem defined by Table 5.6 was run again in PDT with the addition of a 20 day component half life. The explicit Euler, implicit Euler, modified Euler, and Runge-Kutta 4 schemes were used to integrate the equations in time for a total simulation time of 32 days. We applied the adjoint formalist to compute the total derivative of the QOI with respect to the total, scattering, and group-2 fission cross sections, as well as the decay constant and initial component density. These numerical derivatives were compared to analytic derivatives to determine the convergence rate of the schemes.

Figure 5.10 shows the error convergence rates. Each subfigure corresponds to a different Runge-Kutta scheme and shows 11 error convergence lines. The 11 lines are: the error in the QOI itself and the error in predicting the sensitivity of the QOI with respect to 2 total cross sections, 4 scattering cross sections, 2 fission cross sections, the initial component density, and the component decay constant. For this simple problem, many of the errors are equal, and thus their lines are overlaid.

We observe that the implicit and explicit Euler schemes are converging at the predicted first order rate. For some of the parameters, we see that sub-cycling the Bateman equation improves the predictions. We also observe that the modified Euler and RK4 schemes are converging at their expected second and fourth order rates, respectively. These results are evidence that the adjoint of the Runge-Kutta schemes is being applied correctly to integrate the adjoint variables. Experience showed that slight mis-indexing (at the sub-time step level) resulted in suboptimal convergence rates; thus, these tests helped to flush out indexing bugs, as is the purpose of verification exercises. These results also indicate that the correct “snapshot” of the forward solution is being used at each adjoint time step to form the sources in the adjoint equations.

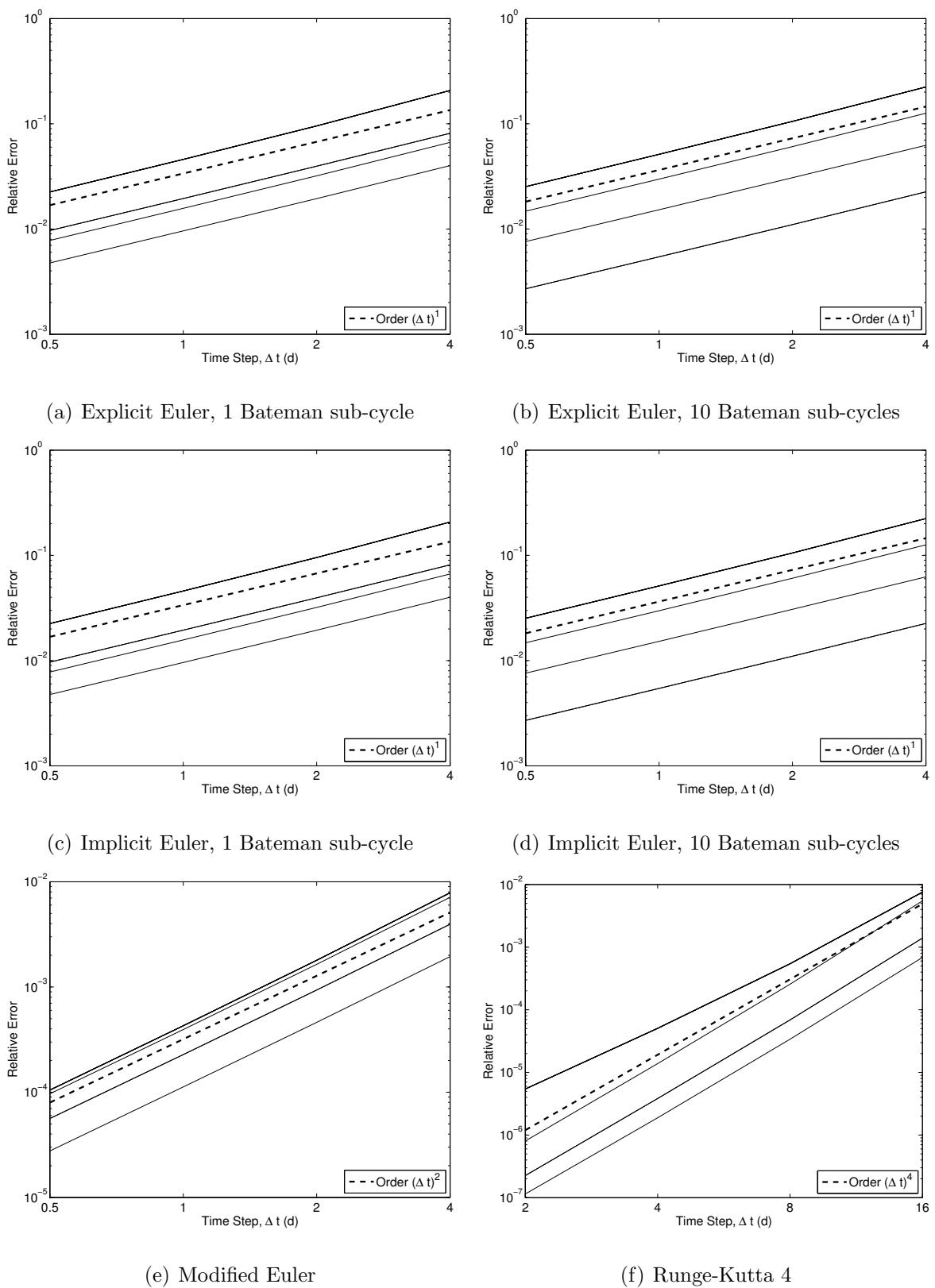


Figure 5.10: Convergence of sensitivity estimates for different Runge Kutta schemes
and Bateman sub-cycle levels

Next we address the accuracy and convergence rate of the adjoint base error estimate for this problem. We will exercise the formalism described in Sec. 3.1.2 using the Heun-Euler embedded Runge-Kutta scheme (see (E.4)), which calls for a first-order scheme solution estimate using the explicit Euler scheme and a second-order solution estimate using the modified Euler scheme.

We begin by analyzing the error made in one time step due to discretization in time. The analytic QOI after one time step T is

$$Q(T) = \frac{V}{N_A} \left[\left(\frac{\kappa}{\lambda} + N_0 \right) \exp(-\lambda T) - \frac{\kappa}{\lambda} \right],$$

which to $\mathcal{O}(T^4)$ expands to

$$Q(T) = \frac{V}{N_A} \left[N_0 - (N_0 \lambda + \kappa) T + \frac{1}{2} (N_0 \lambda^2 + \kappa \lambda) T^2 - \frac{1}{6} (N_0 \lambda^3 + \kappa \lambda^2) T^3 \right] + \mathcal{O}(T^4).$$

The QOI predicted by the explicit Euler rule is

$$Q^{(1)} = \frac{V}{N_A} [N_0 - \kappa T - \lambda N_0 T],$$

where we use the notation superscript- (p) to indicate a p^{th} order estimate. We maintain our definition of error as $\Delta Q^{(p)} = Q - Q^{(p)}$, and find that the true error

made by this scheme is

$$\begin{aligned}
\Delta Q_{\text{true,local}} &= Q(T) - Q^{(1)}(T) \\
&= \frac{V}{N_A} \left[\left(\frac{\kappa}{\lambda} + N_0 \right) \exp(-\lambda T) - \frac{\kappa}{\lambda} \right] - \frac{V}{N_A} [N_0 - \kappa T - \lambda N_0 T] \\
&= \frac{V}{N_A} \left[N_0 - (N_0 \lambda + \kappa) T + \frac{1}{2} (N_0 \lambda^2 + \kappa \lambda) T^2 - \frac{1}{6} (N_0 \lambda^3 + \kappa \lambda^2) T^3 \right] \\
&\quad - \frac{V}{N_A} [N_0 - \kappa T - \lambda N_0 T] + \mathcal{O}(T^4) \\
&= \frac{V}{2N_A} \left[(N_0 \lambda^2 + \kappa \lambda) T^2 - \frac{1}{3} (N_0 \lambda^3 + \kappa \lambda^2) T^3 \right] + \mathcal{O}(T^4). \tag{5.37}
\end{aligned}$$

We'll now apply our formalism to compute an adjoint-based estimate of the time discretization error after a single time step. Inherent in this estimate is an assumption that the algebraic equations are solved exactly; this assumption is valid in this case, because the Bateman equation does not depend on the flux solution. Thus, the only discretization error in the Bateman solution is that due to the time stepping scheme.

Under this assumption, Eq. (3.86) reduces to

$$\Delta Q_{\text{predicted,local}} = - \int_0^T N^\dagger(s) r_1^d(s) ds \tag{5.38}$$

where $r_1^d(s)$ is an estimate of the differential residual, defined by Eq. (3.20). The embedded Runge-Kutta scheme gives information about the integrand of Eq. (5.38) at $t = T$ only; therefore, we choose to integrate this equation using a fully explicit scheme that is consistent with our integration of the adjoint and sensitivity equations:

$$\Delta Q_{\text{predicted,local}} \approx -T N^\dagger(T) r_1^d(T).$$

To compute this residual, we write the exact Bateman equation,

$$\frac{dN}{dt} + \kappa + \lambda N = 0,$$

and the discrete equation that we solve, which leaves some residual relative to the exact equation:

$$\frac{dN^{(1)}}{dt} + \kappa + \lambda N^{(1)} = r_1^d.$$

Subtracting the first expression gives us an expression for r_1^d :

$$\begin{aligned} r_1^d &= \frac{dN^{(1)}}{dt} - \frac{dN}{dt} + \kappa + \lambda N^{(1)} - \kappa - \lambda N \\ &= \frac{N^{(1)} - N_0}{T} - \frac{N - N_0}{T} + \lambda(N^{(1)} - N) \\ &= \left[\frac{1}{T} + \lambda \right] [N^{(1)} - N]. \end{aligned} \tag{5.39}$$

As we do not know the exact solution, N , our strategy is to use a higher-order estimate, $N^{(2)}$, generated by the modified Euler rule, in its place. The second order solution at $t = T$ is

$$N^{(2)} = N_0 - T \left[\kappa + \lambda N_0 - \frac{1}{2} \lambda T (N_0 \lambda + \kappa) \right].$$

Substituting this expression into Eq. (5.39), we have an estimate for the differential

residual at $t = T$:

$$\begin{aligned}
r_1^d(T) &= \left[\frac{1}{T} + \lambda \right] \left[N^{(1)} - N^{(2)} \right] \\
&= \left[\frac{1}{T} + \lambda \right] \left[N_0 - \kappa T - \lambda N_0 T - \left(N_0 - \kappa T - \lambda N_0 T + \frac{1}{2} \lambda T^2 (N_0 \lambda + \kappa) \right) \right] \\
&= -\frac{1}{2} \lambda T \left[1 + \lambda T \right] \left[N_0 \lambda + \kappa \right] \\
&= -\frac{1}{2} \left[N_0 \lambda^2 T + \lambda \kappa T + N_0 \lambda^3 T^2 + \kappa \lambda^2 T^2 \right]. \tag{5.40}
\end{aligned}$$

Using our explicit integration rule, our predicted QOI error estimate is

$$\begin{aligned}
\Delta Q_{\text{predicted,local}} &= T \frac{V}{N_A} \frac{1}{2} \left[N_0 \lambda^2 T + \lambda \kappa T + N_0 \lambda^3 T^2 + \kappa \lambda^2 T^2 \right] \\
&= \frac{V}{2N_A} \left[(N_0 \lambda^2 + \lambda \kappa) T^2 + (N_0 \lambda^3 + \kappa \lambda^2) T^3 \right] \tag{5.41}
\end{aligned}$$

Comparing Eqs. (5.37) and (5.41), we see that local truncation error in the prediction of the QOI error is proportional to T^3 ! Therefore, we would expect the error in the global QOI error estimate to be proportional to T^2 ; that is

$$\epsilon_Q \equiv \Delta Q_{\text{true}} - \Delta Q_{\text{predicted}} \propto T^2$$

This is a surprising result considering that we are using only a first order method to integrate the governing equations.

Also of note is the convergence rate of the *relative* error in predicting the global QOI error; that is

$$\epsilon_{Q,\text{rel}} \equiv \frac{\Delta Q_{\text{true}} - \Delta Q_{\text{predicted}}}{\Delta Q_{\text{true}}}.$$

For this problem, the preceding analysis tells us that $\epsilon_{Q,\text{rel}} \propto T$, indicating that the estimate of the QOI error converges more rapidly than the QOI error itself.

This is important. If the ratio did not decrease as T decreases, the QOI error estimate would not be as effective because the error in the estimate would have the same order of magnitude as the quantity it is trying to describe. The fact that the ratio decreases with T would allow the modeler to perform simple extrapolation calculations to determine the effect of decreasing the step size to improve QOI accuracy.

We test these predicted convergence rates by solving the same single component, infinite medium problem as above with the Heun-Euler embedded Runge-Kutta scheme. We simulated 32 days using 0.125, 0.5, 1, 2, 4, 8, and 16 day time steps. After each run, we compiled the true QOI error (using the analytic QOI expression) and the adjoint-based QOI error prediction. The convergence of ϵ_Q and $\epsilon_{Q,\text{rel}}$ are shown in Figs. 5.11 and 5.12, respectively. Note that we achieve the exact rates of convergence as predicted by the above analysis.

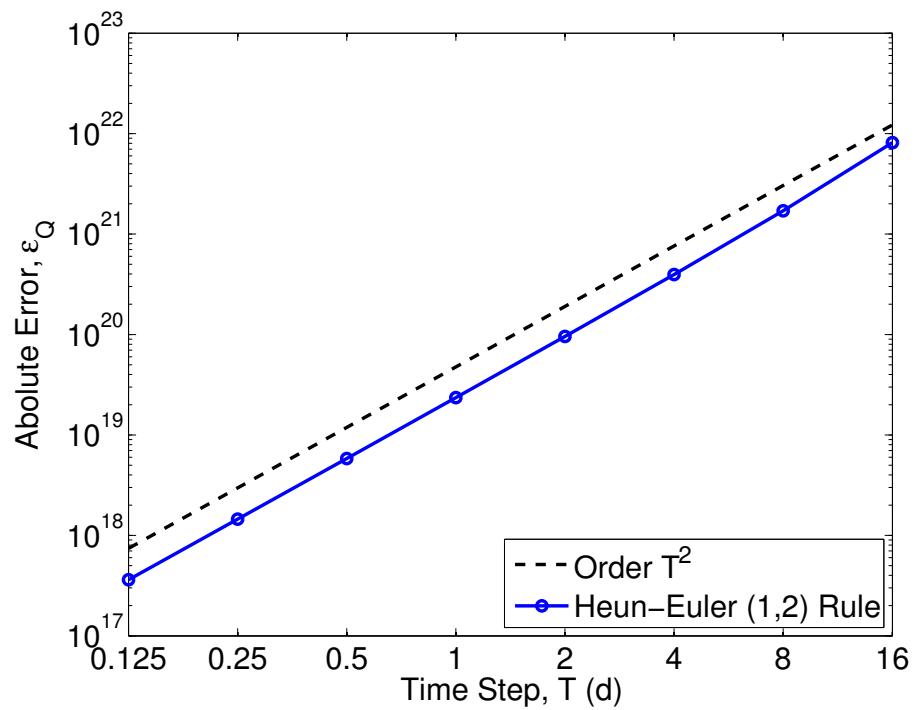


Figure 5.11: The absolute prediction of the error in our QOI due to time discretization is converging to the true error at the expected rate of $\mathcal{O}(T^2)$.

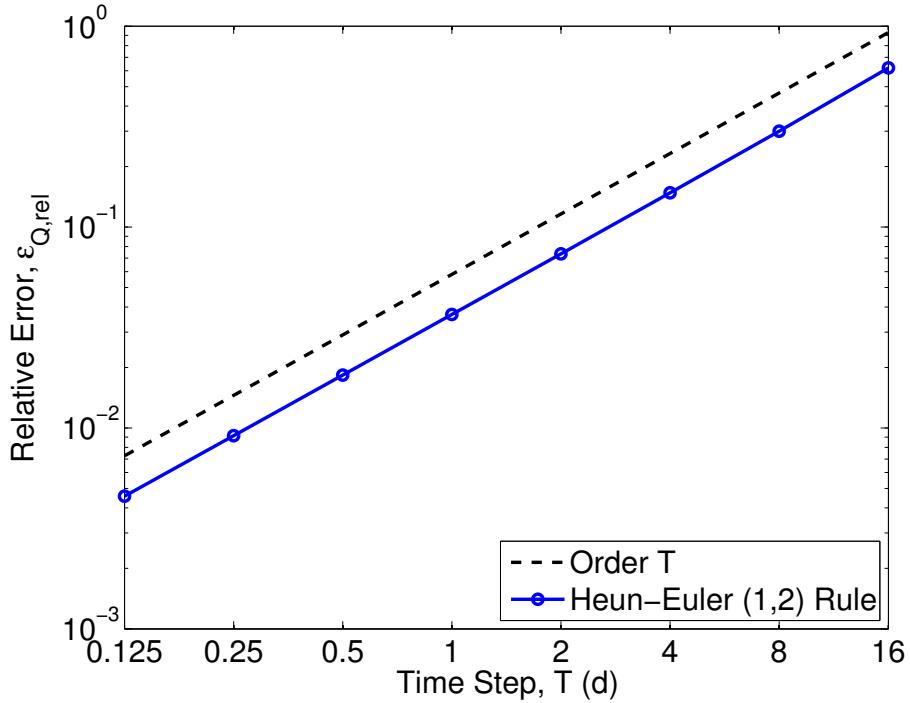


Figure 5.12: The ratio of the accuracy in the QOI error estimate to the accuracy in the QOI itself is decreasing at the expected rate of $\mathcal{O}(T)$.

This is a powerful verification problem. It demonstrates that

1. the forward k-eigenvalue depletion problem is being solved at the correct order of accuracy;
2. the required orthogonality properties, namely

$$\langle S^\dagger, \psi \rangle_{E, \mathcal{D}, \Omega} = \langle \psi^\dagger, G\psi \rangle_{E, \mathcal{D}, \Omega} = 0$$

of the forward and adjoint operators hold;

3. the adjoint formalism leads to the exact discrete derivatives of two different QOIs with respect to a range of parameters;

4. PDT is correctly computing these derivatives;
5. the derivative estimates converge to the analytic derivatives at an optimal rate with respect to time (that is, at the rate of the Runge-Kutta method);
6. PDT is achieving these optimal rates;
7. the adjoint formalism provides an accurate estimate of global QOI time discretization error; and
8. PDT is computing these error estimates correctly, and they are converging at the optimal rates.

This verification problem led to the elimination of a number of software bugs, many related to orthogonality conditions and indexing related to maintaining adjoint consistency. Verification problems for the source-driven case are enumerated in Appendices C.1 and C.2. The first is a walk-through of a very simple pure absorber problem. The second is very similar to the problem discussed here. The combination of these problems provides confidence that the depletion perturbation solver is solving the equations correctly; however, this problem did not test the more complex Bateman dynamics or the effects of spatial variation in the neutron flux on the nuclide densities. The problem described in the following section attempts to address these effects.

5.4 Two Source-Driven Test Problems for Verifying the Accuracy of the Parameter Derivatives

These verification problems are similar to the problem described in Sec. 5.2.2 in that we wish to test the accuracy of the computed parameter sensitivities by comparing to brute force finite difference perturbations. As described before, the sensitivity equation (Eq. (3.52) in the case of the source-driven depletion equations) produces an estimate of the gradient, $\frac{dQ}{dp} \Big|_{p=p_0} \equiv \nabla_p Q_0$, in the differential neighborhood about the modeler-defined parameter vector, p_0 . If computed correctly, this gradient should capture the first order sensitivity of the QOI with respect to perturbations in p_0 . Consider the QOI, $\tilde{Q} \equiv Q(\tilde{p})$ resulting from a perturbed parameter vector, $\tilde{p} = p_0 + \delta p$, written as a Taylor series expansion about the unperturbed QOI:

$$\tilde{Q} = Q_0 + \delta p \cdot \nabla_p Q_0 + \mathcal{O}\left\|\delta p^2\right\|.$$

Define $\delta Q \equiv \tilde{Q} - Q_0$, the QOI perturbation. By rearrangement of the previous expression, we see that the error in a prediction of δQ made by dotting δp with our gradient, $\nabla_p Q_0$, should decrease as the square of the norm of the perturbation:

$$\epsilon_{\delta Q} \equiv \delta Q - \delta p \cdot \nabla_p Q_0 = \mathcal{O}\left\|\delta p^2\right\|.$$

We designed a family of problems to verify this order of accuracy for the depletion perturbation case, and two are presented in this section. Both problems are source driven with two groups and four components. We use the depletion chain and unperturbed cross sections are given in Appendix F.1.

The first problem is a homogeneous, 3D brick of dimensions 30.0cm×30.0cm×10.0cm discretized into a matrix of 8×8×8 spatial cells allocated among 8 processors. The

initial densities $\left[\frac{\text{atom}}{\text{b}-\text{cm}}\right]$ for U-235, U-238, U-239, and Fe-56 are 1.0, 5.33, 1.8, and 1.0 respectively. We use S_8 level-symmetric quadrature in angle. In time, we use the modified Euler Runge-Kutta scheme (see Appendix E) to march through 8 time steps of $5.0\text{e}+05$ seconds each, and the CHECKPOINT_MOMENTS scheme is used to perform the adjoint cross-correlation. A fixed isotropic neutron source of $4.20\text{e}+12$ and $1.20\text{e}+12 \left[\frac{n}{\text{cm}^3-\text{sec}-\text{MeV}}\right]$ in groups 1 and 2 is constant throughout the domain. Finally, the QOI is defined as the number of moles of Fe-56 present in the entire problem at the end of the simulation.

The second problem geometry is pictured in Fig. 5.5 and is discretized into a matrix of $6\times 6\times 6$ spatial cells on one processor. The filler and detector material initial densities are given in Table 5.9. Here we use S_2 level-symmetric quadrature in angle. We march through time using the modified Euler scheme with 40 cycles of $5.0\text{e}+05$ seconds each. The CHECKPOINT_MOMENTS scheme is used to perform the adjoint cross-correlation. The fixed neutron source is $4.20\text{e}+14$ and $1.20\text{e}+14 \left[\frac{n}{\text{cm}^3-\text{sec}-\text{MeV}}\right]$ in groups 1 and 2, respectively, throughout the domain. Here, the QOI is the total interaction rate with U-235 in the detector volume at $t = t_f$.

Table 5.9: Initial densities for the detector response depletion perturbation verification problem

Component	Filler	Detector
U-235	0.0	1.0
U-238	1.8	0.0
U-239	0.0	0.0
Fe-56	1.0	3.0

For each problem, step one of the verification task is to compute $\nabla_p Q_0$ about the unperturbed cross sections given in Appendix F.1. We then perturb the following one-at-a-time perturbations to p_0 :

$$\tilde{p}_{ij} = p_0 + \Delta_j e_i,$$

where $e_i = \langle 0, 0, \dots, 0, 1, 0, \dots, 0, 0 \rangle$ and $\Delta = [.001, .002, .004, .008]$. In other words, we perturb each cross section 4 times, and for each perturbation of each cross section, we compute \tilde{Q}_{ij} . We then make the prediction

$$\tilde{Q}_{ij,\text{pred}} = \Delta_j e_i \cdot \nabla_p Q_0$$

and compute the error

$$\epsilon_{\delta Q,ij} = \tilde{Q}_{ij} - \tilde{Q}_{ij,\text{pred}}.$$

This error should increase by a factor of 4.0 as the perturbation, Δ_j is doubled.

Figure 5.13 shows the convergence of $\epsilon_{\delta Q,ij}$ as a function of the perturbation in the total, fission, nu-bar, and scattering cross sections in the nuclide inventory problem. Figure 5.14 gives the same set of plots for the detector response problem. We give a Δ^2 line on each set of axes for reference. The results show that the implementation achieves the predicted second order convergence rate for these two problems.

These problems help to verify that the depletion and adjoint depletion solvers are interacting properly for a problem with more complex geometry and Bateman dynamics than in previous test problems. These problems include fission, decay, (n,2n) and (n, γ) reactions, a relatively large number of unknowns, a higher burnup of the nuclides, and the additional degree of difficulty involved with using multiple processors. The results are strongly second order, providing further confidence that

the parameter derivatives are being computed correctly.

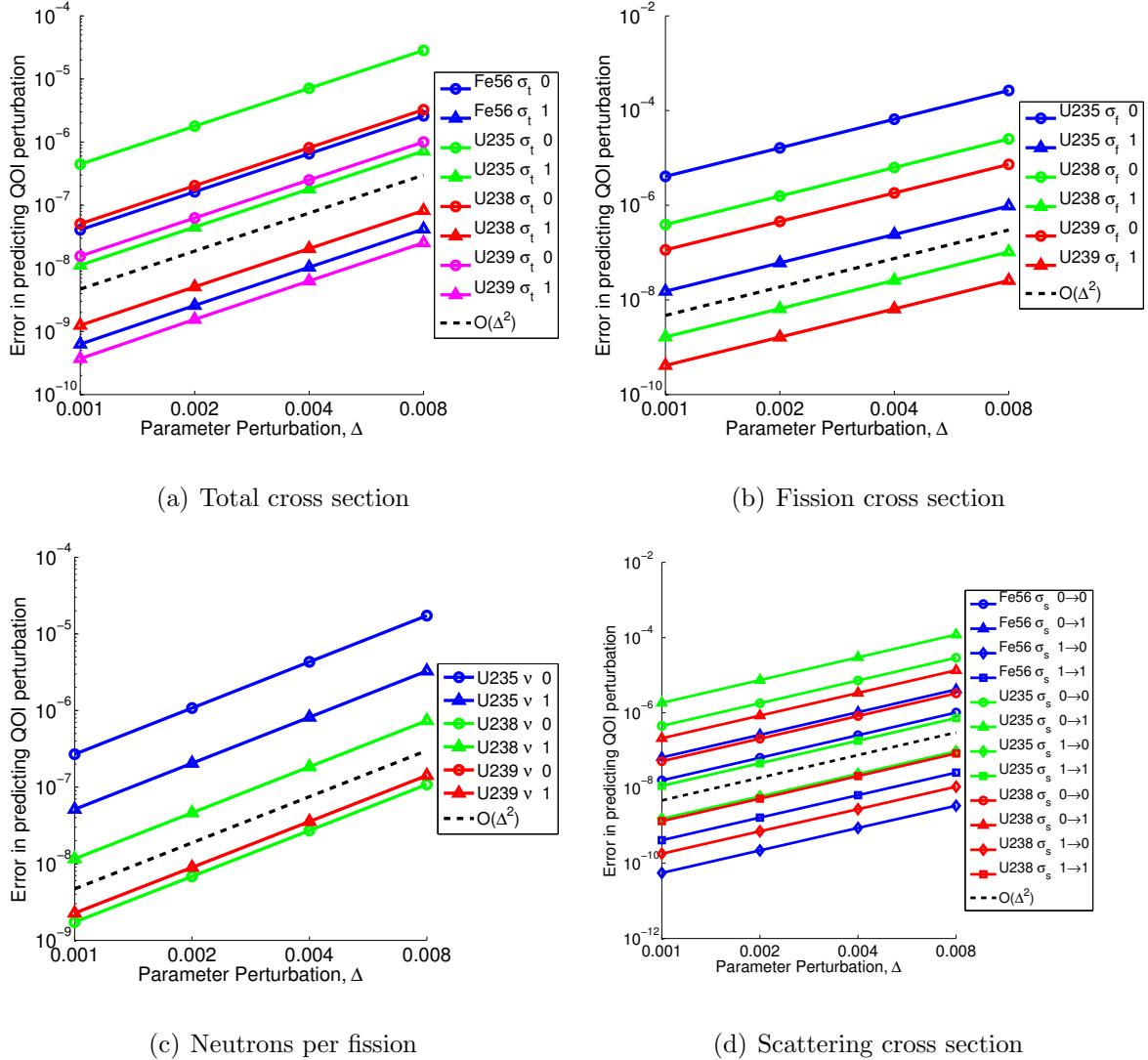


Figure 5.13: The error in predicting δQ , where Q is the terminal nuclide density, decreases as the square of the parameter perturbation.

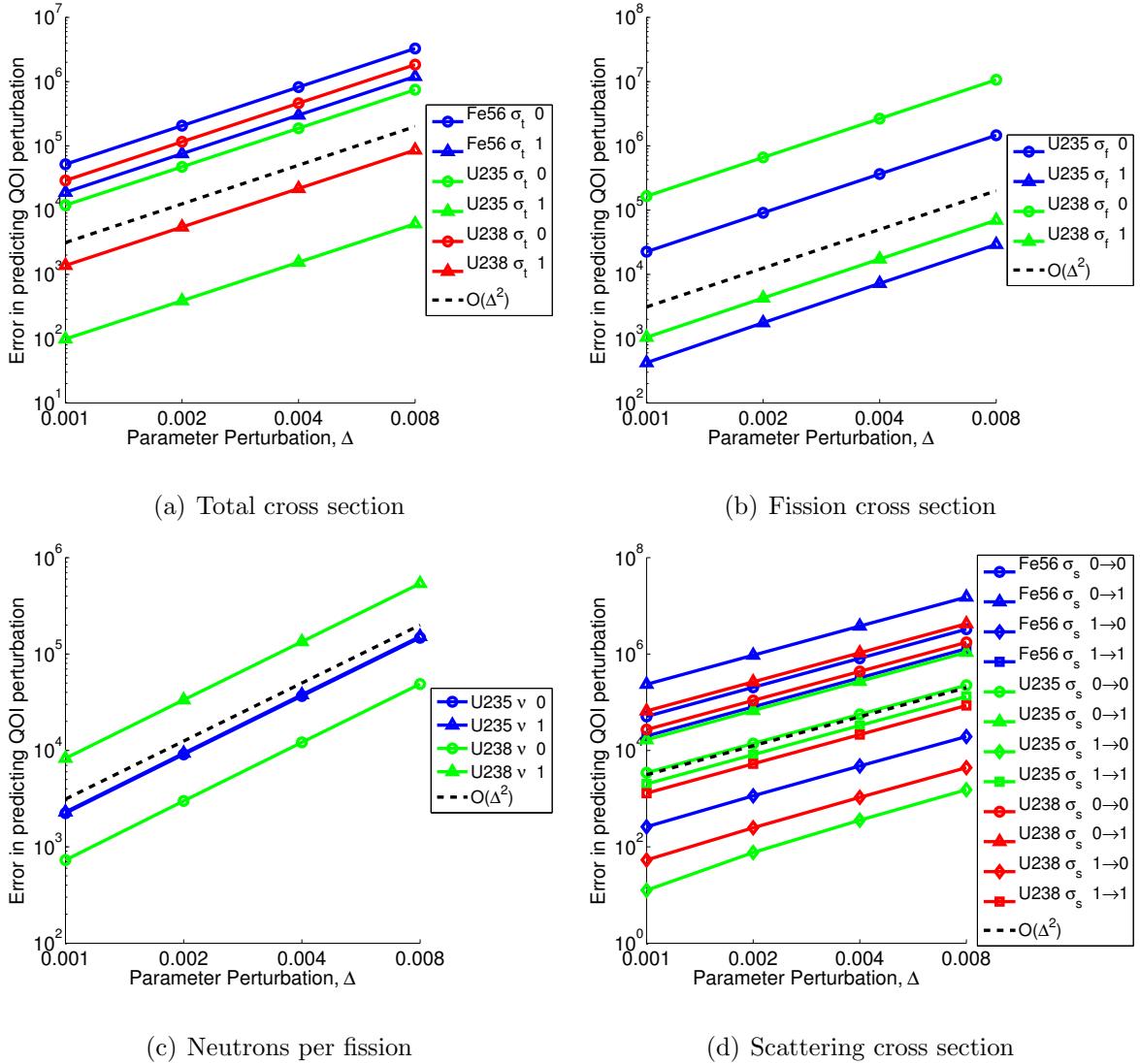


Figure 5.14: The error in predicting δQ , where Q is a model of a detector response, decreases as the square of the parameter perturbation.

6. APPLICATION AND SCALING PROBLEMS

In this section, we present the application of our depletion and adjoint capability to two reactor problems that are of interest to ongoing research by the department and its collaborators. The first is a traveling wave reactor benchmark problem, originally proposed by staff members at TerraPower, LLC. The task is to model a one-dimensional reactor over a series of operational cycles until an equilibrium cycle is reached. We report our results and enhance the benchmark problem by applying a series of UQ studies. The second reactor problem involves the simulation of the Nuclear Science Center (NSC) TRIGA research reactor at Texas A&M University. We are interested in modeling the power history of the reactor and in calibrating uncertain cross sections and other parameters using experimentally measured quantities. In this dissertation, we provide an example NSC depletion calculation and show how the adjoint information can be used to calibrate poison concentrations.

The last section of this section describes a scaling study designed to test the performance of the checkpointing algorithms that were described in Sec. 4.6. We scale a source-driven problem in two measures: the total number of processors and the total number of unknowns per processor. Our conclusion is that the schemes that read, write, and store only the converged transport source moments are computationally advantageous compared to the traditional checkpointing algorithms.

6.1 A One-Dimensional Traveling Wave Reactor Benchmark Problem with Depletion and Uncertainty Quantification

This section describes our approach to and solution of a benchmark problem simulating a one-dimensional traveling wave reactor design. Robert C. Petroski studied the problem in detail in his 2011 dissertation from MIT[56]; months later, with co-authors from TerraPower, LLC, he proposed the simulation as a benchmark problem to the reactor physics community[57].

In the following subsections, we give a complete definition of the benchmark problem and summarize available results from Petroski *et. al.* We then describe our approach to the forward problem and its solution using the PDT code. In the last subsection, we apply our adjoint formalism to the benchmark problem to perform a number of simulated UQ studies.

6.1.1 Problem description and reference results

The breed & burn reactor core is modeled as an infinite slab with 100 inseparable, homogenized assemblies of width 5cm. A constant, average power density of 48 W/cm³ is specified, and the reactor is run over a series of 450 day depletion cycles. Between depletion cycles, a shuffling scheme is applied such that two fresh, or “feed”, assemblies are inserted at each end of the slab and a particular burnt assembly is removed from the interior of the slab. The material composition of the feed assemblies, which corresponds to a roughly 0.3% w/o enriched fuel mixture, is given in Table 6.1.

Table 6.1: Composition of feed material for traveling wave reactor benchmark

Nuclide	Density [$\frac{\text{atom}}{\text{b}-\text{cm}}$]
Uranium-235	7.30e-05
Uranium-238	2.40e-02
Sodium-23	6.52e-03
Iron-56	1.68e-02

After some number of shuffle sequences, the reactor will enter an equilibrium state where each cycle does not differ from the cycle before it. The benchmark task is to reach this equilibrium state for two different shuffling sequences and to report a number of equilibrium cycle reactor performance characteristics. The two shuffling sequences are referred to as inward-convergent shuffling (ICS) and convergent-divergent shuffling (CDS) and are illustrated in Fig. 6.1.

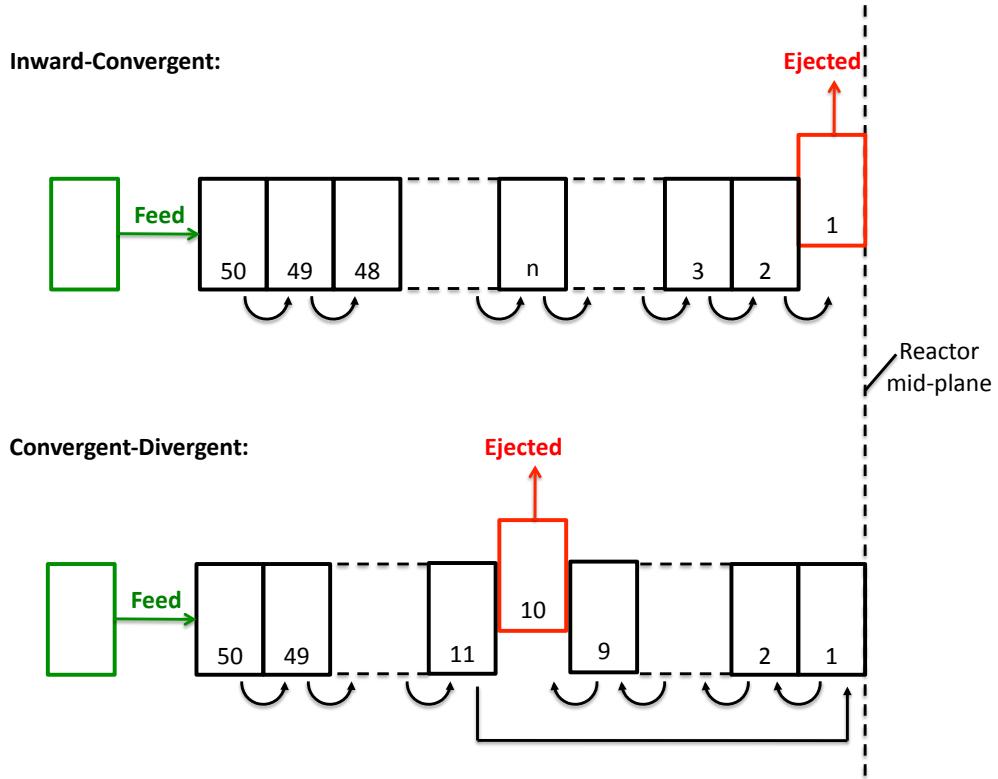


Figure 6.1: Shuffling schemes for the traveling wave reactor problem: note the shuffling is symmetric about the reactor mid-plane.

Reference results are drawn from both the Petroski dissertation and the benchmark specification. The authors indicate they used a modified version of MCNPX to for their results. Figure 6.2 is taken from the benchmark specification [57] and shows the “uncontrolled eigenvalue”, k_{eff} , over the equilibrium cycle for each shuffling strategy. Values for the beginning of equilibrium cycle (BOEC), middle of equilibrium cycle (MOEC), and end of equilibrium cycle (EOEC) eigenvalues are tabulated in the dissertation[56] and are reproduced in Table 6.2.

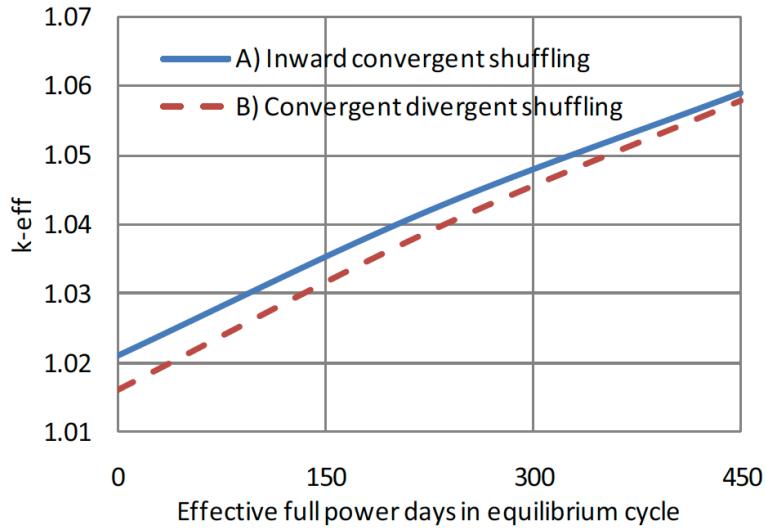


Figure 6.2: Reference k-eigenvalue solutions for the TWR problem.

Petroski also provides BOEC and EOEC curves for burn-up, power density, and total scalar flux. The benchmark solution requires more detailed, tabular data describing the equilibrium cycle, including isotopic edits. We will report this data and compare to available reference solutions in the following subsection.

Table 6.2: Tabulated reference equilibrium cycle eigenvalues for the traveling wave benchmark problem (given as ± 0.001)

Parameter	ICS	CDS
BOEC k_{eff}	1.021	1.016
MOEC k_{eff}	1.042	1.039
EOEC k_{eff}	1.059	1.058
Δk_{eff}	0.038	0.042

6.1.2 Approach and solution using the PDT code

We apply the forward depletion solver in PDT to compute solutions to the benchmark problem. As described in Sec. 2.1, we express the time dependence of the forward problem in quasi-static form, meaning that the Bateman equation is written in time-differential form while the transport solution is solved for a steady-state flux at certain time intervals. We use both the operator splitting technique (see Sec. 4.2.1) and the Runge-Kutta scheme (see Sec. 4.2.2) to solve the quasi-static equations, and we found that the schemes produced consistent results.

For nuclear data, we use a subset of 33 nuclides from the data described in Appendix F.2. These are 33-group, fast spectrum cross sections provided by staff members at Argonne National Laboratory using the MC² deterministic spectrum code. The scattering data contains 4 Legendre moments per nuclide; it also includes energy-per-fission data, but not half-life data, which we gathered directly from ENDF databases.

Of the 33 nuclides we track, 18 compose the actinide chain, which is illustrated by Fig. 6.3. A crucial part of this chain is the Plutonium-239 production process, which occurs via neutron capture in Uranium-238 to produce Uranium-239, followed by beta decay to Neptunium-239 and a second beta decay to Plutonium-239. The half-life of Uranium-239 is just less than 24 minutes. To avoid numerical instabilities that may be caused by this short time scale, we assume than a capture reaction in Uranium-238 produces Neptunium-239 directly, as indicated by the illustration.

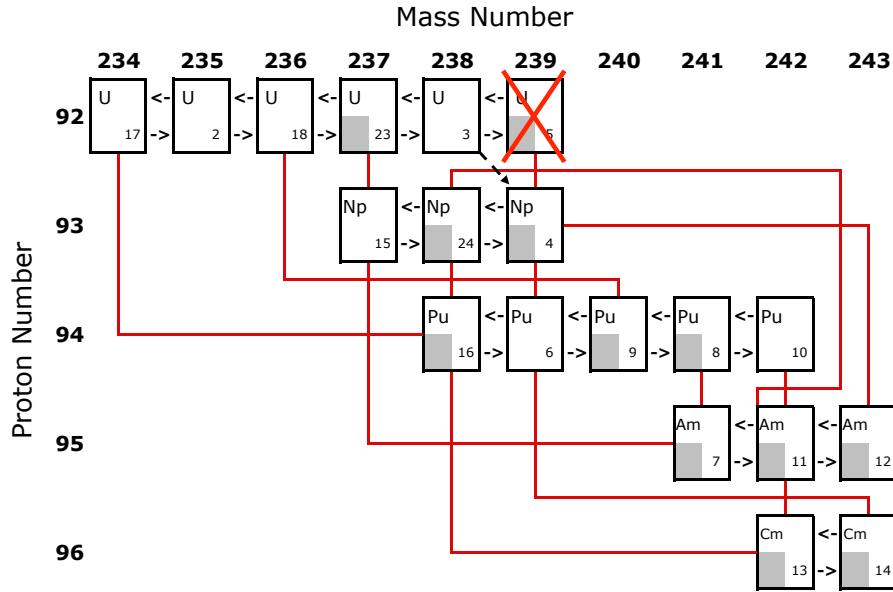


Figure 6.3: Actinide chain for the traveling wave reactor benchmark problem. Grey boxes indicate unstable isotopes; red lines indicate decay parent/child pairs; black lines indicate reaction parent/daughter pairs.

Thirteen of the remaining nuclides are lumped fission products, each of which corresponds to a particular fissionable nuclide. We neglect fission products for ^{242}Cm and ^{243}Cm and do not have fission product data for ^{237}U , ^{238}Np and ^{239}Np . One fission product is produced per fission in the corresponding parent nuclide. The lumped fission products do not decay.

The remaining two nuclides are Sodium-23 and Iron-56. We treat the sodium as a filtered coolant, so its concentration is not allowed to change during the simulation; the iron nuclides are removed via absorption, but we do not model any products of

this reaction.

For each shuffling scheme, we solved the problem to equilibrium state using a variety of initial nuclide compositions. In this section, we show the approach to equilibrium from a near-equilibrium state. Specifics of our solution strategy are given in Table 6.3. Note that PDT does not support 1D geometries; therefore we model this problem using 100 2D spatial cells of dimension $1.0\text{e}+07\text{cm} \times 5.0\text{cm}$.

Table 6.3: Specification of problem and solver settings for equilibrium cycle benchmark solution using PDT. Note: all tolerances are relative, and flux tolerances specify max point-wise, group-wise change in scalar flux.

Parameter	Setting
Runge-Kutta scheme	Implicit Euler
Depletion time step	45 days
Bateman sub-cycle length	12 hours
Fixed-source solver	Source iteration
Fixed-source iterative tolerance	1.0e-04
Eigenvalue/vector solver	Power iteration
Eigenvalue tolerance	1.0e-04
Eigenvector tolerance	1.0e-03
Spatial method	PWLD
Angular quadrature	S_8
Scattering Moments	4 (P3)

We find that increased resolution of each assembly in space does not affect the solution, a result also observed by Petroski. We also find that the solution is insensitive

to quadrature order and scattering order so long as some anisotropy is allowed (we explore this in further detail below). The implicit Euler scheme for time marching solves the Bateman equation implicitly, but it uses an explicit flux solution. The Bateman equations are also sub-cycled between flux solutions. Because the flux solution changes very little over the equilibrium cycle, we find that this is an effective time stepping scheme.

Figure 6.4 shows k_{eff} during the approach to equilibrium for each shuffling scheme as well as a snapshot of the eigenvalue over a few equilibrium cycles of each scheme. The plots give the PDT result in solid line and the reference BOEC, MOEC, and EOEC k_{eff} values from Table 6.2 in dashed lines. It is interesting to compare Figs. 6.4(a) and 6.4(b). The initial density profiles from which these jobs were launched were identical. The ICS scheme has a fairly smooth slope to equilibrium, while the CDS scheme has a nonlinear path.

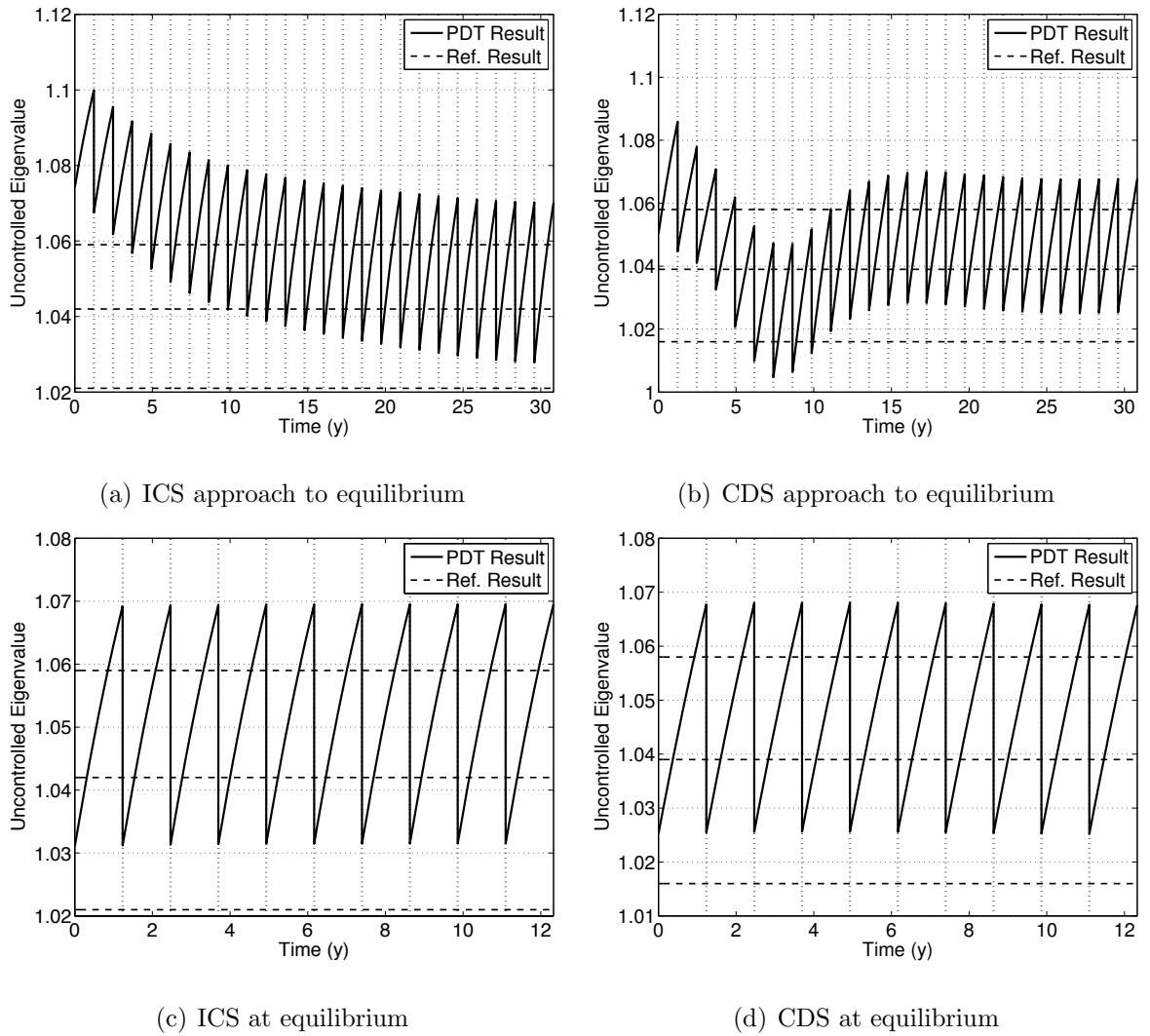


Figure 6.4: Approach to equilibrium and repeated equilibrium cycles for both the inward convergent and convergent divergent shuffling schemes

Table 6.4 appends the PDT results to Table 6.2 for tabular comparison. Figure 6.5 compares the equilibrium cycle PDT solution to the reference solution in the format of Fig. 6.2.

Table 6.4: Comparison of tabulated equilibrium cycle eigenvalues for the traveling wave benchmark problem (± 0.001)

Parameter	Reference	PDT	Reference	PDT
	ICS	ICS	CDS	CDS
BOEC k_{eff}	1.021	1.031	1.016	1.027
MOEC k_{eff}	1.042	1.052	1.039	1.050
EOEC k_{eff}	1.059	1.070	1.058	1.069
Δk_{eff}	0.038	0.039	0.042	0.042

We find that our eigenvalue results over-predict the reference results by about 1.0%. The trends in the PDT curves (e.g. concavity, relative difference between schemes) are highly consistent with the reference curves. Because we found that further refinement in space and angle did not affect the PDT solutions, we believe the shift in our predictions is likely due to different treatment of cross sections, nuclide compositions, and energy group structure. We do not know which nuclides and data libraries produced the reference solutions; we only know that the reference solution authors used a version of the MCNPX code.

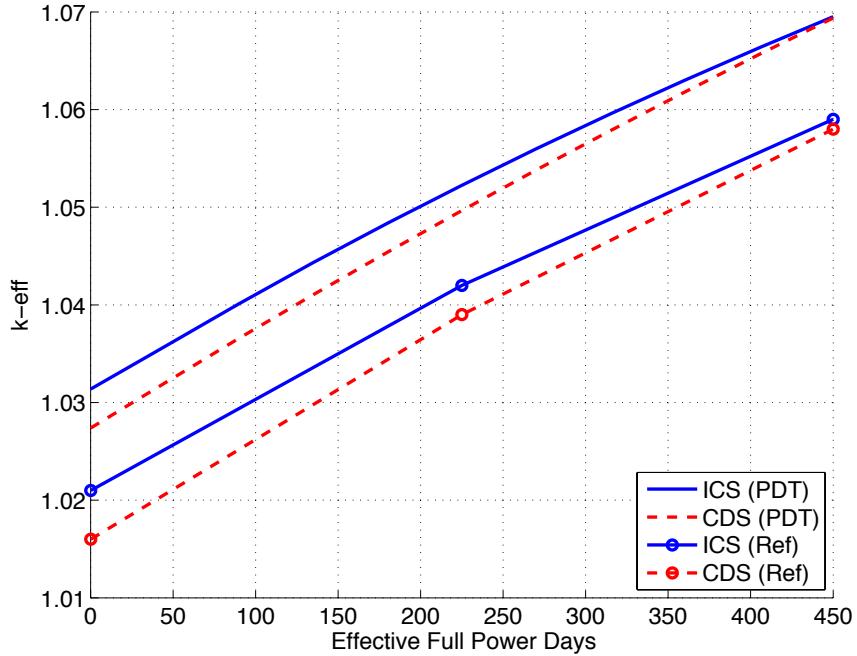


Figure 6.5: Comparison of reference and PDT uncontrolled eigenvalue solutions over a single equilibrium cycle. Note that the reference curves are interpolated between the available data at BOEC, MOEC, and EOEC.

The benchmark specification requests a large amount of equilibrium cycle data in addition to what has been shown thus far. First, the authors request a set of three BOEC, MOEC, and EOEC global neutronics parameters for each shuffling scheme, which we report in Table 6.5. The global neutronics parameters are the eigenvalue k_{eff} , global average number of neutrons per fission $\bar{\nu}$, and global average energy released per fission, \bar{Q} . The latter two parameters are defined as:

$$\bar{\nu} = \frac{\langle \psi \nu \Sigma_f \rangle_{E,\mathcal{D},\Omega}}{\langle \psi \Sigma_f \rangle_{E,\mathcal{D},\Omega}}$$

$$\bar{Q} = \frac{\langle E_f \psi \Sigma_f \rangle_{E,\mathcal{D},\Omega}}{\langle \psi \Sigma_f \rangle_{E,\mathcal{D},\Omega}}$$

Table 6.5: Global neutronics parameters computed with PDT for the TWR benchmark problem (± 0.001)

Parameter	Inward-Convergent			Convergent-Divergent		
	BOEC	MOEC	EOEC	BOEC	MOEC	EOEC
k_{eff}	1.031	1.052	1.070	1.027	1.050	1.069
$\bar{\nu}$	2.897	2.899	2.901	2.896	2.898	2.900
\bar{Q} (MeV)	198.6	198.6	198.6	198.6	198.6	198.6

In addition to the global neutronics parameters, the benchmark requests cell-wise powers, absorption and fission edits, and isotopic concentrations. Table 6.6 reports these data for a select set of assemblies (numbered from the center of the slab outward) and the ICS scheme. FIMA, or fissions per initial metal atom, is a burnup metric and is defined as the total number of fissions in a cell divided by the initial number of heavy metal atoms. The last column is the ratio of total neutron absorption to total neutron production via fission in the cell.

Table 6.6: Benchmark results for select assemblies in the traveling wave reactor problem

Cell	EOEC Power Dens. (W/cc)	EOEC Rel. Fiss. Rate	EOEC FIMA (%)	Ratio: $\frac{\text{Tot Abs}}{\text{Tot. Prod}}$
1	4.234e+02	1.000e+00	12.18	0.783
2	4.036e+02	9.533e-01	10.03	0.7899
3	3.669e+02	8.668e-01	7.99	0.806
4	3.170e+02	7.490e-01	6.12	0.835
5	2.595e+02	6.133e-01	4.51	0.879
6	2.011e+02	4.754e-01	3.20	0.945
7	1.476e+02	3.490e-01	2.18	1.039
8	1.029e+02	2.433e-01	1.43	1.172
9	6.842e+01	1.618e-01	0.90	1.355
10	4.363e+01	1.032e-01	0.56	1.606

The electronic version of this dissertation provides the data in Table 6.6 for all cells as well as more detailed isotopic data. For some such data, we are able to compare the PDT results to the reference results. Figure 6.6 compares the PDT equilibrium cycle spatial profiles for burnup, energy-integrated flux, and power density to those given for the reference solution in Petroski's dissertation[56]. The reference solution is only given for the ICS scheme and is only available in graphical format, making direct number-to-number comparisons unavailable.

A possible conclusion from these plots is that the PDT solution is burning in or retaining less fissionable material than the reference solution. If we assume that the energy per fission data for the heavy atoms in each problem are equal, then PDT is

requiring more fissions per initial atom to produce the same amount of energy over the lifetime of a particular assembly. If the energy per fission is the same, then this means that PDT has, on average, less available fissionable material.

The current PDT model truncates the actinide chain. For example, a capture reaction in ^{242}Pu , ^{243}Am , or ^{243}Cm results in a loss of the nucleus and no production of the “A+” child. Similarly, ($n,2n$) reactions may result in the loss of a nucleus but no “A” child. If we tracked more actinides in a future iteration, we would likely see more agreement in the burnup, flux, and power density profiles because more fissionable material would be available for energy production.

We also looked at the BOEC and EOEC nuclide density profiles in the slab and compared their characteristics for each shuffling scheme. Fig. 6.7 gives these profiles for ^{235}U and Fig. 6.8 gives the profiles for the nuclides involved in the ^{239}Pu production chain.

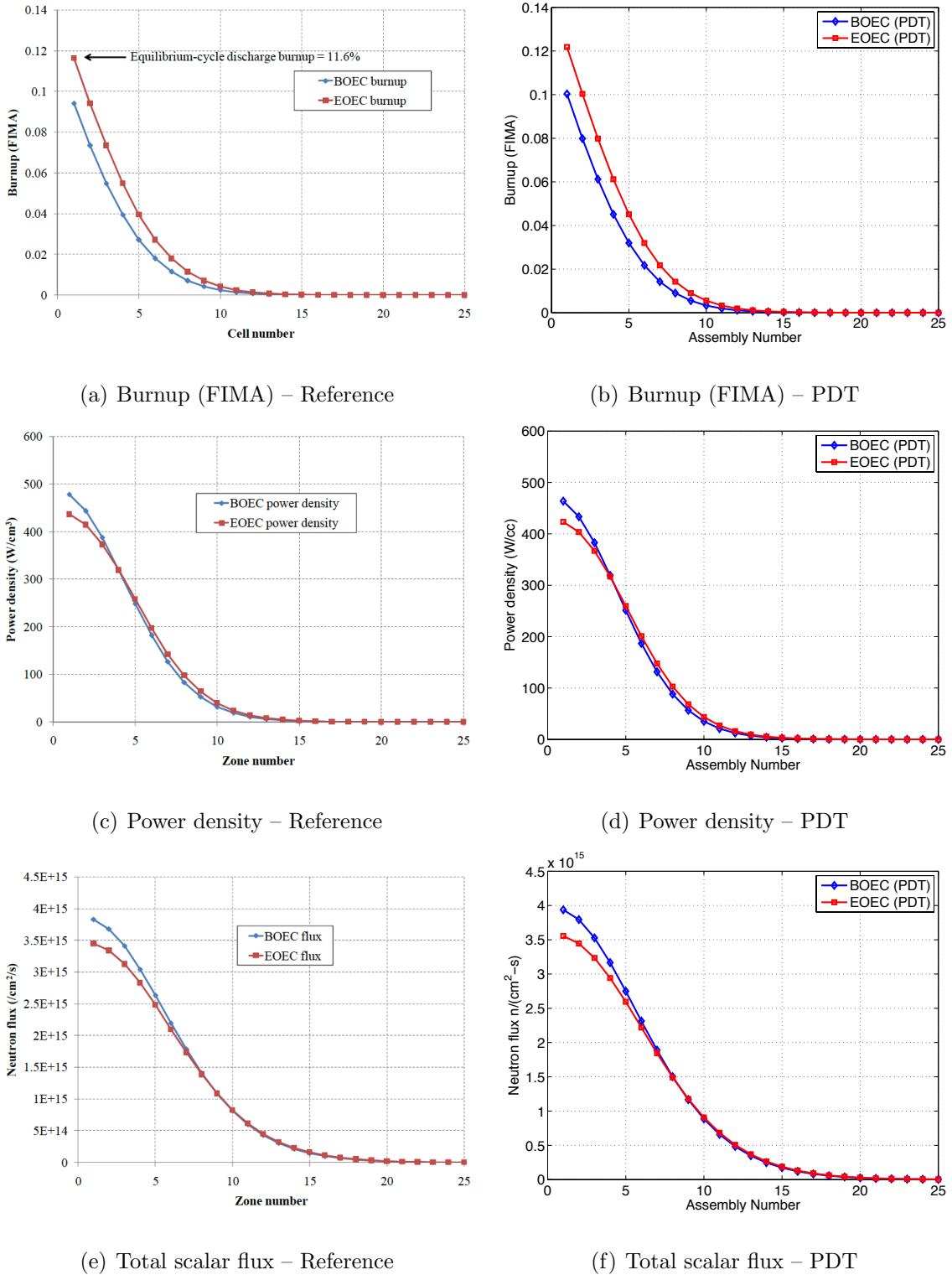


Figure 6.6: Comparison of PDT and reference profiles of burnup, flux, and power density in the equilibrium cycle.

We also looked at the BOEC and EOEC nuclide density profiles in the slab and compared their characteristics for each shuffling scheme. Fig. 6.7 gives these profiles for ^{235}U and Fig. 6.8 gives the profiles for the nuclides involved in the ^{239}Pu production chain.

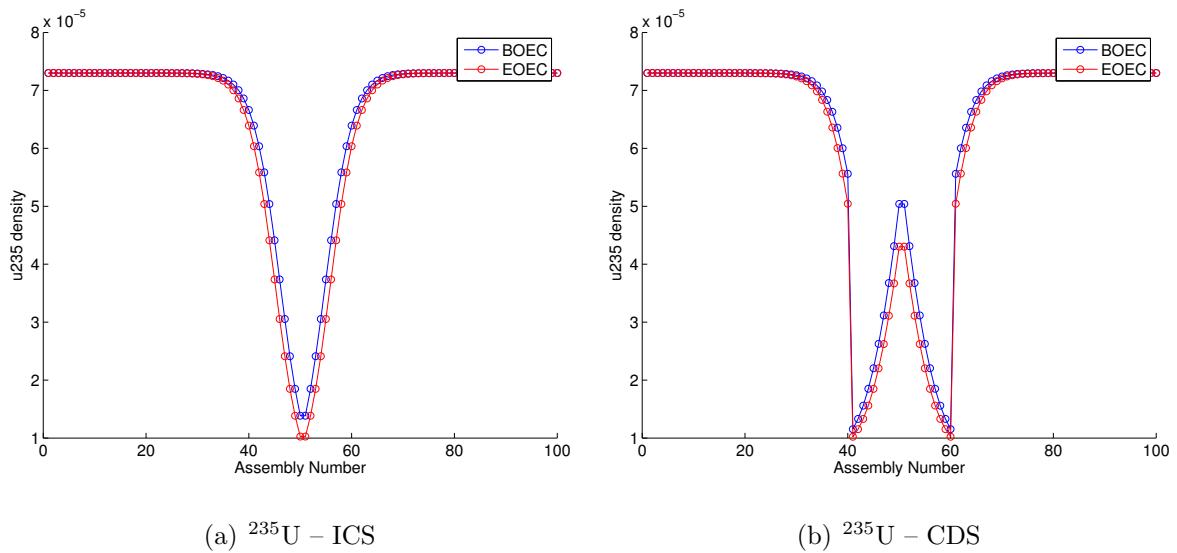


Figure 6.7: BOEC and EOEC density profiles for ^{235}U

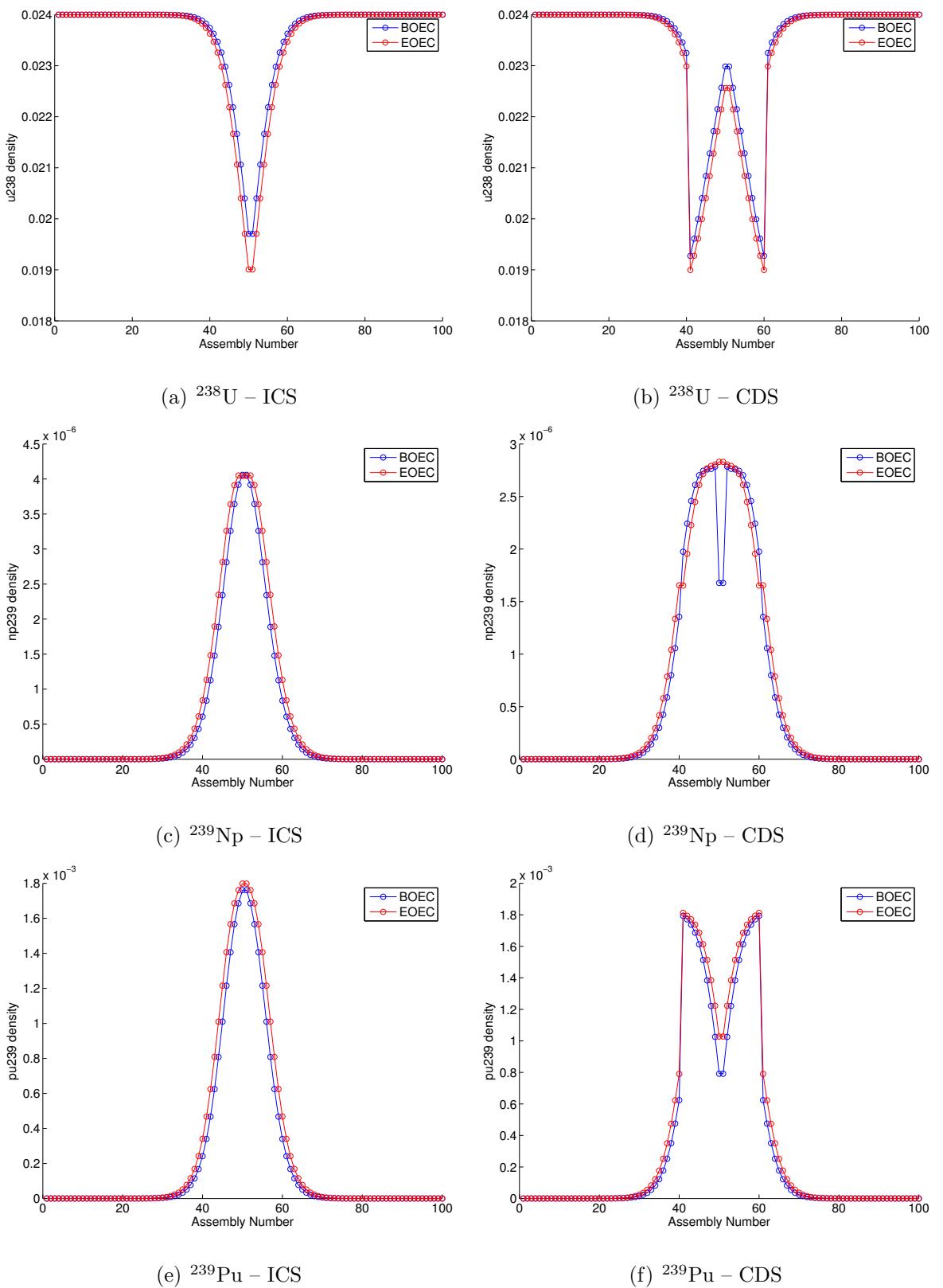


Figure 6.8: BOEC and EOEC density profiles nuclides in the ^{239}Pu production chain.

Finally, we tested the sensitivity of our results to quadrature order and number of Legendre moments used to represent the scattering source. Figure 6.9 compares the uncontrolled equilibrium-cycle ICS eigenvalue for 5 different cases: P0-S₄, P1-S₄, P1-S₂, P2-S₄, and P3-S₈. We find that the eigenvalue is not sensitive to the degree of anisotropy modeled as long as some anisotropy is accounted for (i.e. P1 is enough). Also, among the anisotropic cases, the eigenvalue is insensitive to the quadrature order.

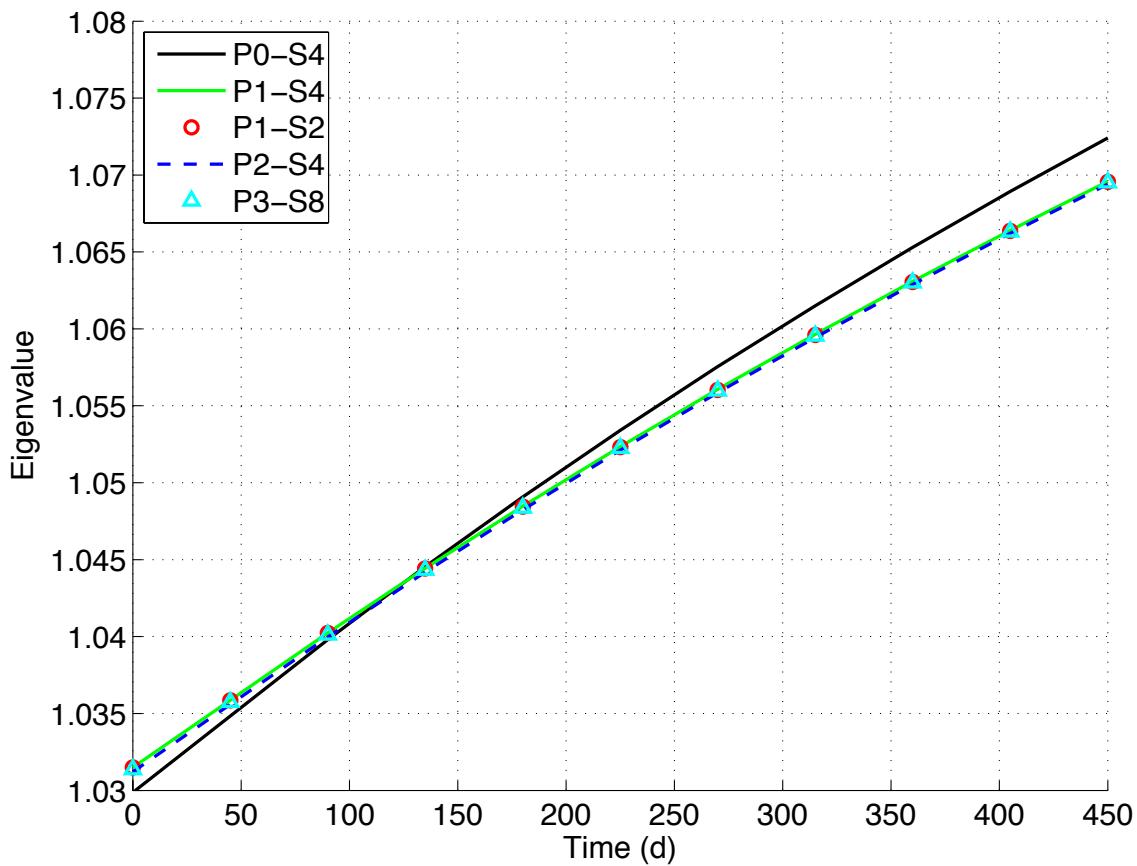


Figure 6.9: Comparison of equilibrium-cycle uncontrolled eigenvalue for 5 different angular and anisotropy models

6.1.3 An equilibrium-cycle uncertainty quantification study

We applied our adjoint capabilities to the benchmark problem in order to learn more about the parameters that are driving the equilibrium cycle behavior. We performed a series of tests, each targeting a different UQ-related question. The first test targets the ^{239}Pu production and utilization in the equilibrium cycle, which is important because our analysis shows that nearly 75% of fissions in the reactor at the EOEC are occurring in this isotope. We concentrate on the ICS shuffling scheme and define a QOI as the total reaction rate in ^{239}Pu at EOEC in the two cells that are to be ejected, or

$$Q = \int_{245\text{cm}}^{255\text{cm}} dx \int dE \int d\Omega \sigma_{t,239}(E) N_{239}(x) \psi(x, E, \Omega).$$

This QOI captures both the Plutonium concentration and neutron flux at the center of the slab just before a shuffle occurs. Using our PDT simulation, we find the nominal value of this QOI is

$$Q_{\text{nominal}} = 7.658e + 21.$$

A first task is to determine which microscopic cross sections are most important in driving the QOI. We do this by running a single adjoint calculation over the nominal ICS equilibrium cycle. This gives us the sensitivity of our QOI with respect to all cross sections and other select nuclear data that we provide to the code. If we make the simple assumptions that all cross sections are independent and known to within a particular percent, then we can rank the importance of the cross sections using the

following sensitivity measure[39, 46]:

$$S_p = p \frac{dQ}{dp}.$$

This measure indicates that important parameters are those with a combination of large influence on the QOI and a large uncertainty (which we have assumed is proportional to p). We certainly note that the assumption that cross sections are independent is completely inadequate for a realistic uncertainty study; instead, correlation matrices must be considered in order to simulate physically viable perturbations in a nuclide's cross sections. We simply use this metric as an example for the simulated UQ study that follows.

We chose to focus on the total and fission cross sections of ^{235}U , ^{238}U , and ^{239}Pu for this particular test. We considered the (n, γ) cross section for ^{238}U as well, but our analysis found that its importance factor was about an order of magnitude lower than the ^{238}U fission cross section. Figure 6.10 shows the importance, normalized by Q_{nominal} , of the total and fission cross-section parameters to this particular QOI as a function of group number.

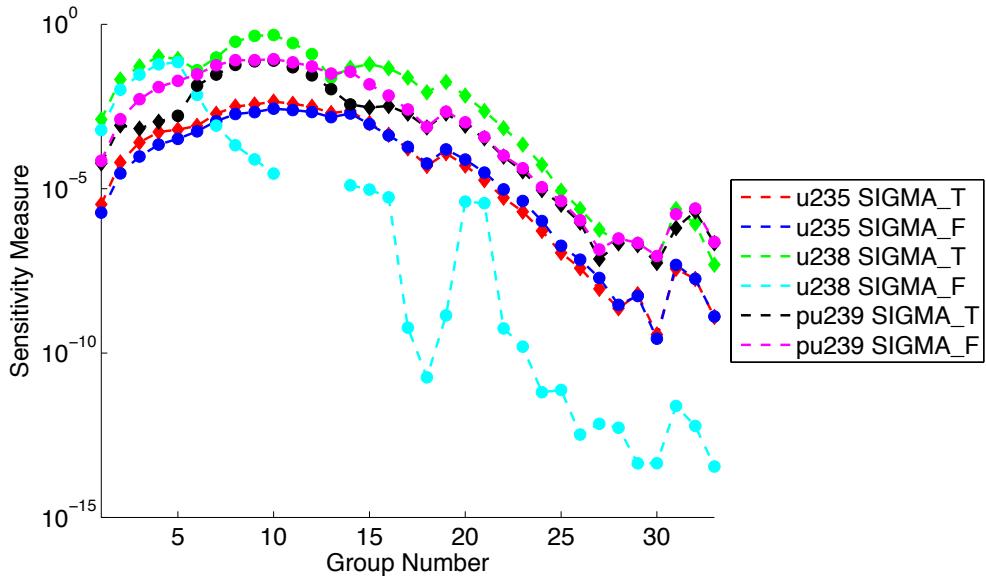


Figure 6.10: Normalized parameter sensitivity measure for the ^{239}Pu reaction rate QOI. Note: diamond markers indicate positive sensitivities, and circle markers indicate negative sensitivities.

An immediate observation from Fig. 6.10 is that the maximum sensitivity occurs in the higher-energy groups, which coincides with the fast flux spectrum in this traveling wave reactor. The total cross section in ^{238}U seems to have the largest potential to drive variation in the QOI, but the fission cross section is only important at the highest neutron energy levels. The ^{235}U cross sections are not as important throughout the energy spectrum, and the Plutonium cross sections are mostly in between.

To simulate a UQ study, we used this plot to identify 18 cross sections, 6 for each nuclide, as parameters of interest. They are listed in Table 6.7. We assumed that each cross section may vary within a uniform distribution with range $\pm 3\%$ about its nominal value. We sampled this 18 dimensional space using a 180 point Latin-hypercube design, which is a scheme used for random stratified sampling[58].

Table 6.7: Parameters chosen for ^{239}Pu reaction-rate UQ study

Uranium-235		Uranium-238		Plutonium-239	
Cross section	Group	Cross section	Group	Cross section	Group
Total	9	Total	4	Total	9
Total	10	Total	10	Total	10
Total	11	Total	15	Total	15
Fission	10	Fission	3	Fission	8
Fission	11	Fission	4	Fission	9
Fission	14	Fission	5	Fission	10

For each of the 180 sampled points, we ran a single equilibrium-cycle simulation using perturbed values for the 18 parameters. We then used the adjoint-based derivatives to compute an estimate for the perturbed QOI as follows:

$$Q_{\text{pred}} = Q_{\text{nominal}} + \sum_{i=1}^{18} \delta p_i \frac{dQ}{dp_i}.$$

Figure 6.11 compares the predicted QOI to the actual QOI for each of the 180 runs. Perfect predictions would lie on the dashed diagonal line. The average predictive error ($\sim 4.6\text{e+19}$) is less than 1% of the QOI value for these 180 runs, which we find encouraging given that the predictions are simply linear extrapolations about the nominal equilibrium cycle.

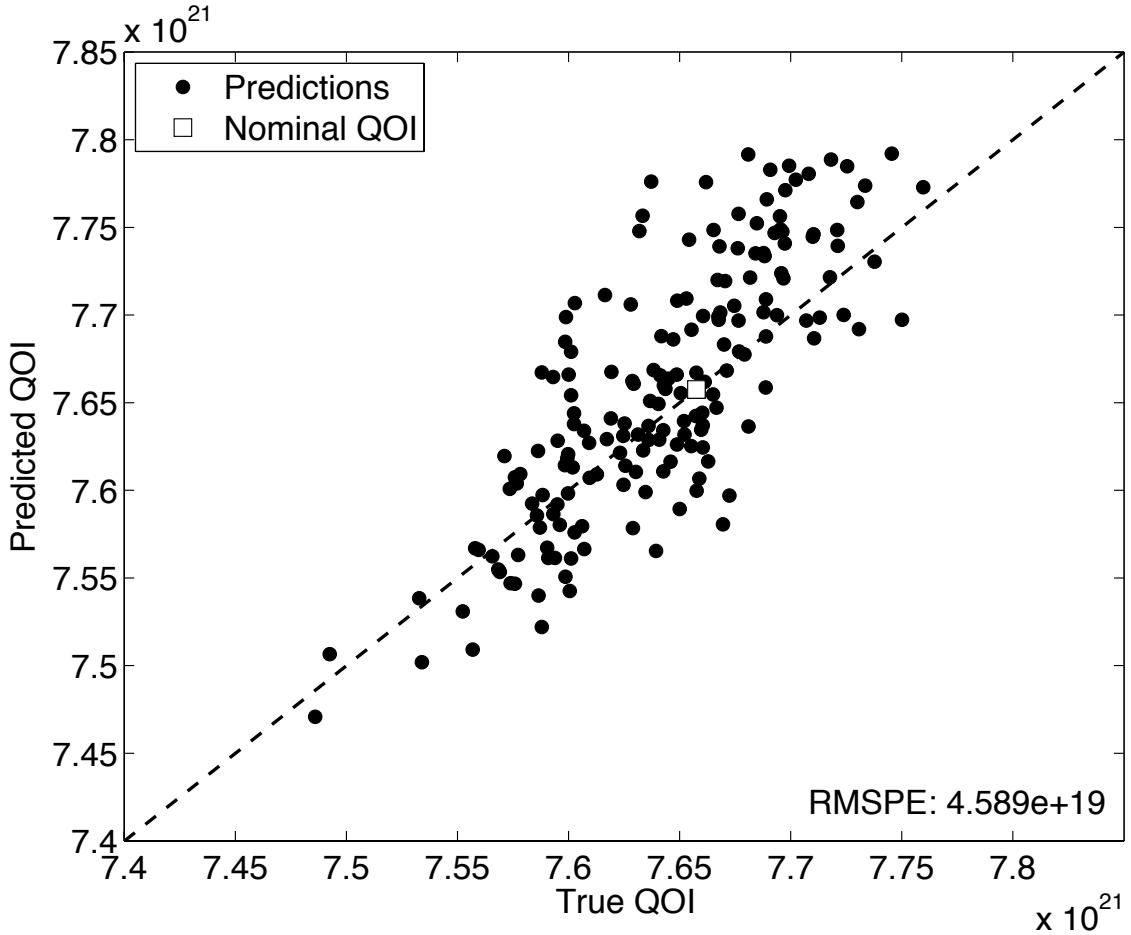


Figure 6.11: Predicted vs actual QOI for the traveling wave reactor UQ study

We also applied our error estimation technique to this simulation. Using the Heun-Euler embedded Runge-Kutta scheme, we estimated a residual due to time-discretization at each time-step during the forward simulation. We showed in Sec. 3.1.2 that integration of these residuals against the adjoint variable results in a global time-discretization error estimate.

Using a 5 day time-step and a solver tolerance of 0.001%, our analysis predicts a global absolute time-discretization error in this QOI of $1.8e+17$ or a relative error of 0.0023%. We compared this to a Richardson extrapolation estimate[16], which

requires a second forward simulation using 2.5 day time-steps. This error estimate was $1.8e+18$, or a relative error of 0.0145%, which is about 6 times larger than our adjoint-based estimate. Both estimates are small, which we expected given the quasi-equilibrium nature of the simulation.

The difference between the estimates is less than one order of magnitude. In the context of error estimation, this is a very small difference, as the purpose of an error estimator is usually to give the modeler an idea of how many digits to trust in an answer. For this problem, the estimators are telling us that the error due to time discretization is on the order of the solver tolerance, or that time-discretization is not a driving source of error.

Perhaps more important for this dissertation, however, is the cost of this error estimate. Let N be the number of k-eigenvalue power iterations required to solve the forward problem using 5 day time steps. The total cost of the adjoint simulation is $3N + 1$, where N come from the forward solve, $2N$ come from the adjoint solve (a homogeneous and inhomogeneous solve at each time step), and 1 comes from the terminal condition calculation. The total cost of the Richardson estimate is $3N - 1$, where N come from the initial 5-day time step run, and $2N - 1$ come from the 2.5-day time step run. Thus, to leading order, the error estimate costs $3N$ transport solves via both methods. The adjoint method, however, comes with the full gradient of the QOI with respect to all the parameters in the problem, information which is totally absent from the Richardson extrapolation.

This simulated UQ study illustrates the wealth of information that one adjoint solve can bring to a modeler. We provided a measure for identifying important parameters for the particular QOI and used this measure to reduce the stochastic dimension down to a manageable size. We then performed sampling in this reduced dimension and showed that adjoint-based predictions of perturbed QOIs agreed with

brute force QOI perturbations to within 1%. This would allow a modeler to produce QOI distributions based on input sensitivities without sampling the forward simulator. Finally, we produced an estimate for global error due to time discretization, which to leading order requires the same computational cost as a Richardson extrapolation and agreed with Richardson to within an order of magnitude.

6.1.4 A multi-cycle uncertainty quantification study

We also used our adjoint capability to characterize the effects of perturbations after several cycle/shuffling sequences. Suppose a proliferation-related QOI is the amount of ^{235}U that is ejected from the reactor as a result of shuffling after each equilibrium cycle. For the nominal reactor configuration, we compute that this QOI is about 399kg. Compare this to the 2850kg inserted per shuffle via the 0.3 w/o enrichment fuel mixture.

Suppose that we wish to compute the sensitivity of this QOI with respect to the enrichment of the feed fuel mixture. This could help a designer determine the appropriate decrease in the enrichment level to reduce the ejected material inventory below a target value. We use a series of adjoint calculations to compute this sensitivity, described as follows.

The initial adjoint calculation is similar to the other adjoint calculations reported in this dissertation. The QOI is defined as the inventory of ^{235}U in the two center assemblies (those that are ejected in the ICS scheme). The terminal adjoint nuclide densities are computed using this definition (see Sec. 3.6.1), and the adjoint equations are integrated to BOEC using the forward equilibrium cycle solution as the reference forward solution. The result of this adjoint calculation is all of the usual QOI sensitivities. In particular, the BOEC ^{235}U and ^{238}U adjoint densities in the outer-most assemblies give the sensitivity of the QOI with respect to the initial feed

concentration of these nuclides after a single cycle.

We are interested, however, in the effect of changing the feed concentration over several cycles. To achieve this, we must perform an adjoint shuffle, which is simply the reverse of the forward shuffle, and repeat the adjoint calculation. In other words, the adjoint densities of the second adjoint calculation are initialized via the reverse shuffling of the assemblies. The adjoint densities in the newly introduced assemblies, which are the inner-most assemblies for the ICS scheme, are initialized to zero. These adjoint variables are then integrated back to BOEC, again linearized about the forward equilibrium cycle solution. Now, the adjoint densities for ^{235}U and ^{238}U correspond to the effect of perturbing the nuclide densities and running two cycles.

Using this pattern, the cumulative effect of perturbing the feed concentration is computed as

$$\frac{dQOI}{dN_{235}} = \sum_{c=1}^{\#cycles} N_{235,c,BOEC}^\dagger,$$

where $N_{235,c}^\dagger$ is the adjoint density corresponding to ^{235}U in the discarded assemblies after the adjoint shuffle. A similar expression exists for ^{238}U .

If this procedure is carried out over C cycles, the total cumulative effect of perturbing the feed concentration can be estimated. We note that as more and more cycles are computed, the validity of linearizing about the nominal equilibrium cycle solution decreases because the perturbed solution is moving further and further from the nominal equilibrium solution. We tested this capability by perturbing the initial feed concentration to 0.25 w/o enrichment. Figure 6.12 shows the predicted and actual perturbation in the ^{235}U inventory resulting from the perturbation in the ^{235}U and ^{238}U feed concentrations.

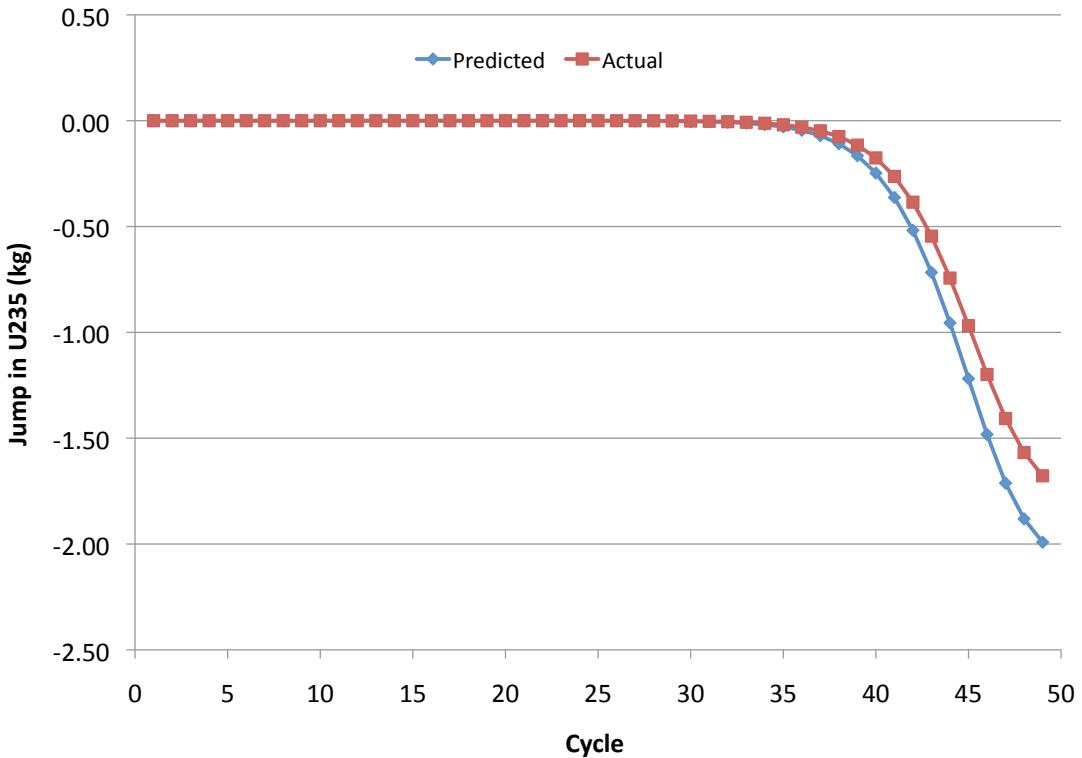


Figure 6.12: Predicted vs actual QOI perturbation for the multi-cycle traveling wave reactor sensitivity test

We observe that it takes about 35 cycles for the perturbed densities to reach the active part of the reactor and impose an appreciable effect on the QOI. The adjoint-based prediction is over-predicting the jump in the QOI, but only by about 10%. The curves stop at 49 cycles. This is because the 50th cycle is when the perturbed density actually reaches the QOI region, representing a relatively large jump (about 300kg) in the QOI. This jump was predicted to within 20% accuracy.

Sensitivity with respect to other parameters can also be estimated in the multi-cycle setting. In this case, the sensitivity with respect to a parameter (e.g. a cross section) is simply the integration of the sensitivity equation over each cycle, or the sum of the sensitivities of each cycle. This provides a straightforward method for

propagating input uncertainties to the multi-cycle QOI. Again, the advantage of the adjoint approach is that these uncertainties can be propagated for any number of parameters without the need to perturb and re-run a large number of forward runs.

6.2 Steady-State and Depletion Simulations of the Nuclear Science Center Research Reactor

This section describes a series of scoping calculations for simulating the research reactor and Texas A&M’s Nuclear Science Center. Long term goals include beginning of life (BOL) calculations, where we will predict critical rod heights and other reactor characteristics, as well as depletion calculations, where we will use the operational history to deplete the core to its present-day isotopic composition. Once a satisfactory model of the reactor core is available, we plan to simulate research experiments and make predictions about future experimental outcomes. We are also interested in the calibration of a Zirconium-Hydride scattering cross-section model, a task which should make heavy use of the sensitivity information produced by the adjoint formalism.

At the time of this dissertation, many of the capabilities and data required for these simulations are under development or just coming online. These include non-orthogonal “reactor” grids, machinery for producing self-shielded cross sections using the Dragon code[59], and the acquisition of reactor power history and other operational data. Therefore, the purpose of the efforts described in this section is to demonstrate and apply the depletion and adjoint capabilities on NSC-like problems in order to facilitate the larger-scale calculations in the future.

6.2.1 NSC beginning-of-life sensitivity calculations

The first task in developing the NSC model is to accurately simulate the BOL configuration. The isotopic composition of the fresh fuel is known to within manufacturing tolerance, and the initial critical core geometry is available in operational records. Comparison of PDT results to the available operational data will allow us to characterize the effects of our discretization choices and nuclear data on the accuracy

of our predictions.

As an example, one important BOL prediction is the critical control rod height. Once a 3D model of the NSC core is available, this will be determined by iteratively adjusting the control rod height until k_{eff} is unity (within some predictive tolerance). Although a manual criticality search is straightforward for this case, it may be more efficient to use the adjoint formalism to produce derivatives and Newton's method to compute the next guess for the critical rod height. This can be achieved by treating the poison and fuel nuclide densities as parameters to which the criticality QOI is sensitive.

To simulate this process, we developed a 3 by 3 pin NSC model using PDT's reactor grids. The grid and material layout are pictured in Fig. 6.13. Material 1, the red material, is representative of the BOL fuel composition. Material 2, shown in green in the figure, is regular water (H_2O) with dissolved Boron-10, a neutron poison. The composition of the fuel and coolant materials are given in Tables 6.8 and 6.9 respectively. We used PDT's power iteration solver to find that the reactivity, $\rho = \frac{k_{\text{eff}} - 1}{k_{\text{eff}}}$, of this system with no dissolved Boron is -1.8063. We ran with angular quadratures of S_2 and S_{12} , and the relative, point-wise eigenvalue and scalar eigenvector tolerances were 1.0e-05.

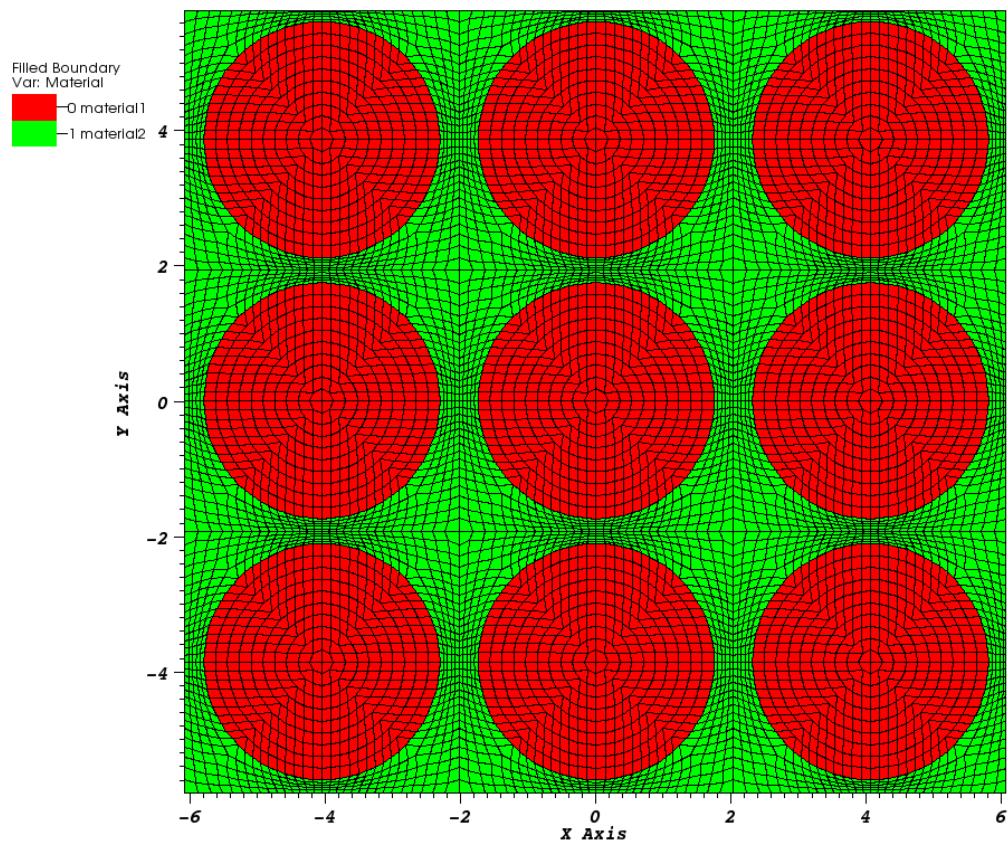


Figure 6.13: Grid and material layout for 9-pin BOL calculation.

Table 6.8: BOL NSC fuel isotopic composition (density units: $\left[\frac{\text{atom}}{\text{b}-\text{cm}} \right]$)

Nuclide	Density	Nuclide	Density
^{234}U	7.11545e-06	^{96}Zr	8.99461e-04
^{235}U	1.08295e-03	^{12}C	1.77837e-03
^{236}U	6.23970e-06	^{174}Hf	3.1224e-09
^{238}U	4.30106e-03	^{176}Hf	1.00341e-07
^{166}Er	7.67971e-05	^{177}Hf	3.58614e-07
^{167}Er	5.27339e-05	^{178}Hf	5.26125e-07
^{90}Zr	1.65276e-02	^{179}Hf	2.62687e-07
^{91}Zr	3.60427e-03	^{180}Hf	6.76521e-07
^{92}Zr	5.50920e-03	^1H	4.89201e-02
^{94}Zr	5.58308e-03		

Table 6.9: BOL NSC coolant isotopic composition (density units: $\left[\frac{\text{atom}}{\text{b}-\text{cm}} \right]$)

Nuclide	Density
^1H	6.6691e-02
^{16}O	3.3346e-02
^{10}B	variable

Suppose our task is to find the boron concentration, N_B , such that the reactivity is $-4.0865 \pm 1.0\text{e-}04$. We happen to know that this reactivity occurs at a concentration of $4.0\text{e-}04 \left[\frac{\text{atom}}{\text{b}-\text{cm}} \right]$. Suppose further that our initial guess is $1.0\text{e-}04 \left[\frac{\text{atom}}{\text{b}-\text{cm}} \right]$. After a single iteration, we would find that the reactivity is -2.6459 , indicating that we need

to increase the ^{10}B concentration. To compute an estimate for the correct increase, we could either (a) iteratively guess-and-check, or (b) compute $\frac{d\rho}{dp}$, where p is the dissolved boron nuclide density, and use Newton's method to compute the update. The latter option requires an adjoint calculation, but should provide the quadratic convergence to the target density.

We pursue the adjoint option. As described in Sec. 3.6, the required adjoint calculation for this QOI is simply the steady-state, homogeneous adjoint k-eigenvalue problem. Upon solving the forward k-eigenvalue problem

$$H\psi - \frac{1}{k_{\text{eff}}}G\psi = 0,$$

and the adjoint k-eigenvalue problem

$$H^\dagger\psi^\dagger - \frac{1}{k_{\text{eff}}}G^\dagger\psi^\dagger,$$

the sensitivity with respect to N_B is

$$\frac{d\rho}{dN_B} = -\left\langle \psi^\dagger, \frac{\partial}{\partial N_B} [H\psi] \right\rangle_{E,\mathcal{D},\Omega}.$$

This calculation gives a gradient with respect to the Boron-10 concentration in *each* spatial cell containing coolant. Because we model a dissolved absorber, we assume that its concentration is the same in each water cell, and thus the total derivative of the QOI with respect to a change in the absorber concentration is the sum of the derivatives in each of these cells.

The problem was run twice: once with S_2 discretization in angle and once with S_{12} discretization in angle. Figure 6.14 shows the spatial variation of the gradient with respect N_B in the S_2 case, and Fig. 6.15 gives the gradient for the S_{12} case.

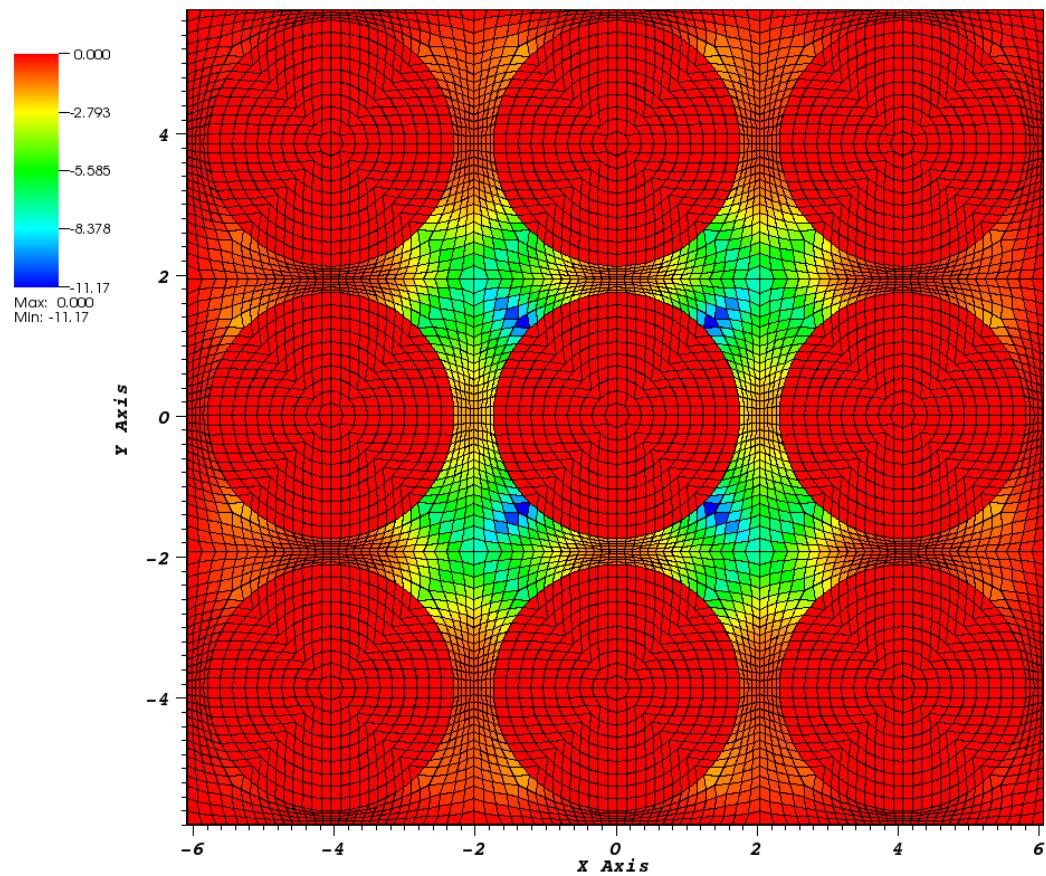


Figure 6.14: Spatial variation of gradient of BOL reactivity with respect to ^{10}B concentration (S_2).

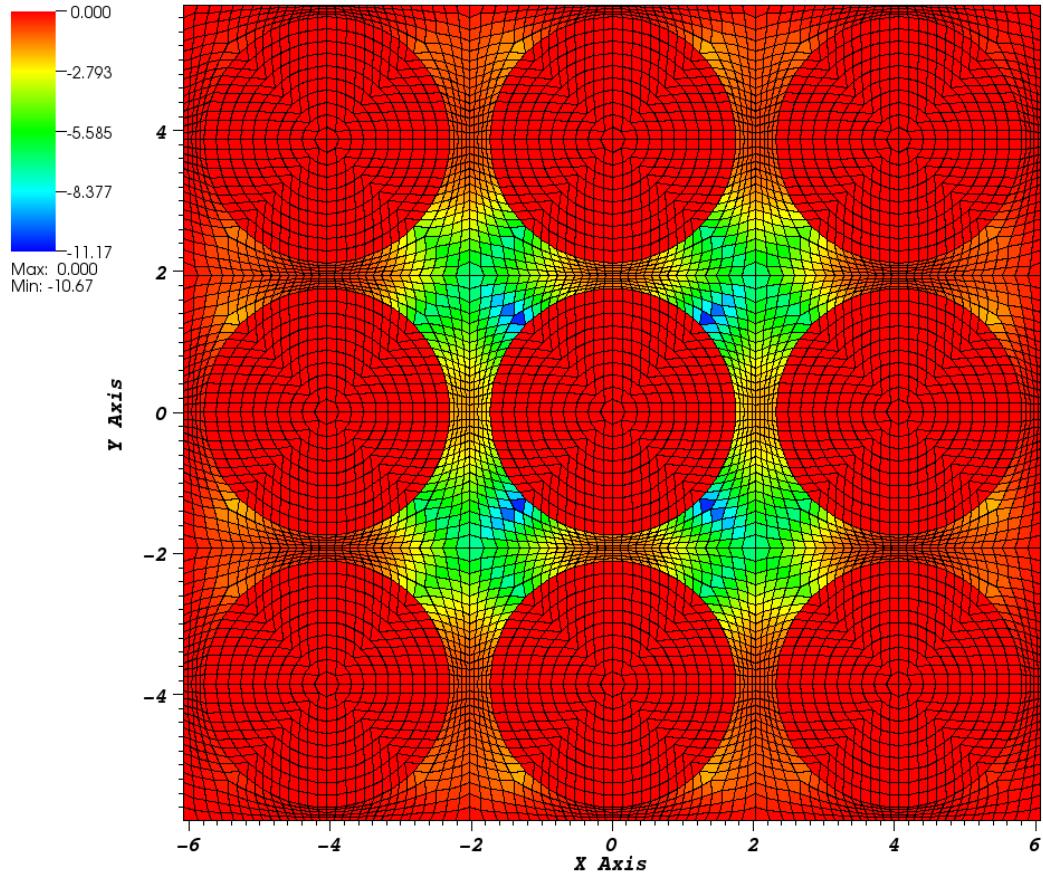


Figure 6.15: Spatial variation of gradient of BOL reactivity with respect to ^{10}B concentration (S_{12}).

The strongest sensitivity occurs on the diagonal transport paths between the center and corner assemblies. This result is consistent with the thermal flux shape, which peaks in these areas (see Fig. 6.16), because ^{10}B is a thermal absorber. At first glance, it appears that the S_2 case is exhibiting ray effects because of the coarse angular discretization. These characteristics persist, however, in the S_{12} case.

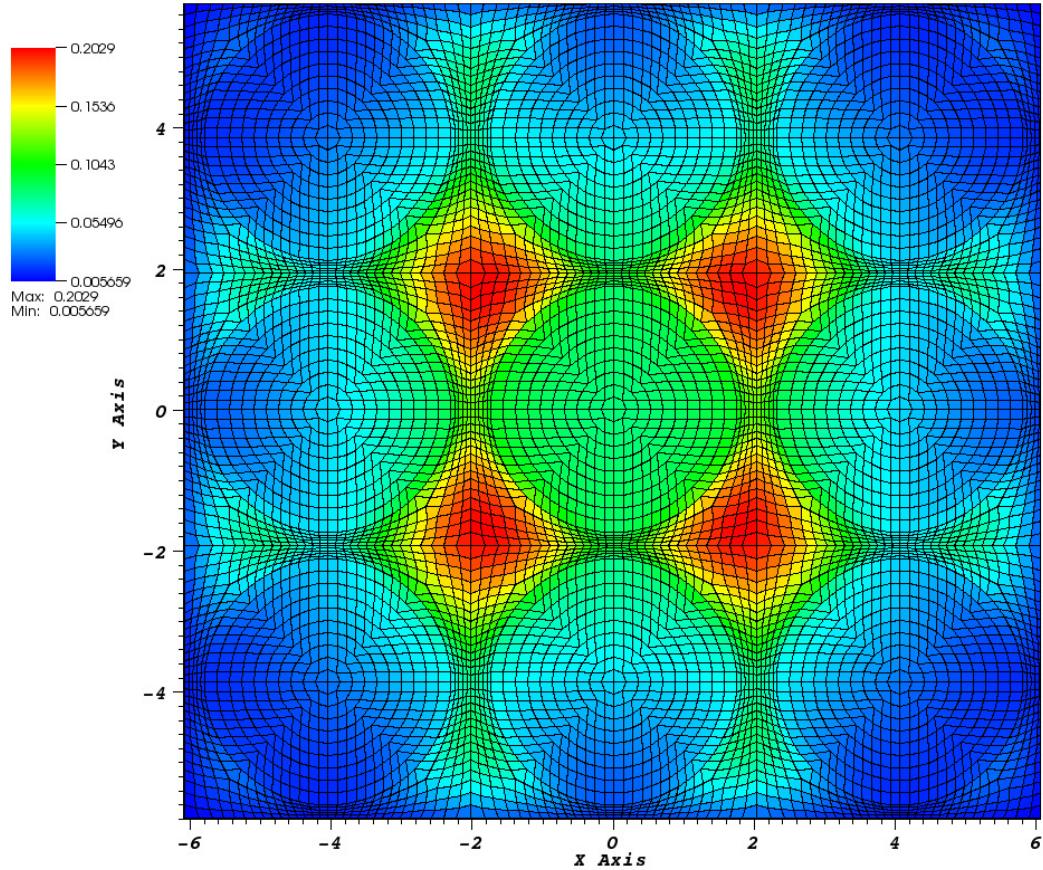


Figure 6.16: Spatial variation of thermal flux at initial ^{10}B concentration (S_{12}).

We use a simple Newton's method to compute the next estimate of the target Boron density. Define $f = \rho(N_B) - \rho_T$, where ρ_T is the target reactivity. The iterative procedure is simply

$$N_B^{\ell+1} = N_B^\ell - \frac{f}{f'} \Big|_{N_B^\ell},$$

where $f' = \frac{df}{dN_B} = \frac{d\rho}{dN_B}$. Table 6.10 summarizes the procedure for each case. Both tests required 4 iterations (or 7 k-eigenvalue solves, 4 forward and 3 adjoint) to find the target poison concentration.

Table 6.10: Summary of Newton iterations towards target Boron density for NSC BOL problem

Iterate, ℓ	Case	N_B^ℓ	$\rho(N_B^\ell)$	$(\rho(N_B^\ell) - \rho_T)/\rho_T$	$N_B^{\ell+1} - N_B^\ell$
1	S ₂	1.00e-04	-2.6459	-3.5e-01	2.0693e-04
	S ₁₂	1.00e-04	-2.6297	-3.6e-01	2.1043e-04
2	S ₂	3.0693e-04	-3.7397	-8.5e-02	8.5029e-05
	S ₁₂	3.1043e-04	-3.6989	-9.5e-02	1.0207e-04
3	S ₂	3.9196e-04	-4.0589	-6.8e-03	7.9965e-06
	S ₁₂	4.1250e-04	-4.0473	-9.6e-03	1.2757e-05
4	S ₂	3.9995e-04	-4.0863	-4.9e-05	n/a
	S ₁₂	4.2526e-04	-4.0861	-9.8e-05	n/a

This example illustrates an application of the adjoint-based gradient calculations for calibrating BOL poison concentrations. The calibration required 7 k-eigenvalue calculations to achieve the desired criticality level within 1.0e-04. It is possible but unlikely that a purely forward, guess-and-check k-search would achieve this result using the same or less k-eigenvalue solves. Moreover, the Newton's method formalism allows for the process to be carried out automatically, using scripts, eliminating the need to manually update the guesses and launch the jobs.

The real BOL calibration task will involve control rod heights instead of soluble poison concentrations. The NSC control rods are “fuel followed”, meaning that a fuel pin follows the control rod up through the core as it is withdrawn. Therefore, the spatial cells through which the rod is moving will need to contain both the control rod materials and the fuel materials, and the material density updates will need to be constrained by what we actually know about the fuel. Nonetheless, a procedure

similar to that outlined in this subsection should be adopted and will likely result in less time and effort in finding the BOL critical rod heights.

6.2.2 NSC depletion calculations

In this section we report results from an example depletion calculation using a 5 by 5 pin-cell model of the NSC core. Each of the 25 pins is modeled as fuel, and the space between the fuel pins is modeled as pure water. Table 6.11 gives the nuclides present in the fuel material and their respective initial densities (note: this list includes all nuclides in Table 6.8, plus nuclides that whose densities will grow as a part of the depletion process).

Table 6.11: Initial fuel material composition for NSC depletion calculation (density units: $\left[\frac{\text{atom}}{\text{b}-\text{cm}} \right]$)

Nuclide	Density	Nuclide	Density	Nuclide	Density
^{234}U	7.11545e-06	^{241}Pu	0.0	^{92}Zr	5.50920e-03
^{235}U	1.08295e-03	^{241}Pu	0.0	^{94}Zr	5.58308e-03
^{236}U	6.23970e-06	^{241}Am	0.0	^{96}Zr	8.99461e-04
^{237}U	0.0	^{242}Am	0.0	^{12}C	1.77837e-03
^{238}U	4.30106e-03	^{243}Am	0.0	^{174}Hf	3.1224e-09
^{237}Np	0.0	^{135}Xe	0.0	^{176}Hf	1.00341e-07
^{238}Np	0.0	^{149}Sm	0.0	^{177}Hf	3.58614e-07
^{239}Np	0.0	^{166}Er	7.67971e-05	^{178}Hf	5.26125e-07
^{238}Pu	0.0	^{167}Er	5.27339e-05	^{179}Hf	2.62687e-07
^{239}Pu	0.0	^{90}Zr	1.65276e-02	^{180}Hf	6.76521e-07
^{240}Pu	0.0	^{91}Zr	3.60427e-03	^1H	4.89201e-02

The actinide chain is exactly the same as that for the traveling wave reactor benchmark (see Fig. 6.3) except that the chain is truncated after Americium. Instead of lumped fission products, we model just two fission products explicitly: Xenon-135 and Samarium-149. All cross sections were produced using the DRAGON code, and half-lives, fission yields, and energy per fission data were obtained from a combination of ENDF-VII data and the data used for the traveling wave reactor benchmark.

The problem is a k-eigenvalue problem. The flux is normalized at a per-pin power level of 6.5 kW, which roughly corresponds to a 62-pin NSC core operating at 400kW. The simulation is one year. The time stepping scheme is implicit Euler with 5-day broad time steps and 0.25 day Bateman sub-cycles.

Figure 6.17 shows the eigenvalue of the system over the first year of operation. We see a very clear initial drop in the eigenvalue, which we attribute to the burn-in of the Xenon and Samarium fission products. This begs the question of whether a 5-day time step (with quarter-day Bateman sub-cycles) is adequate at the beginning of the simulation. We address this below.

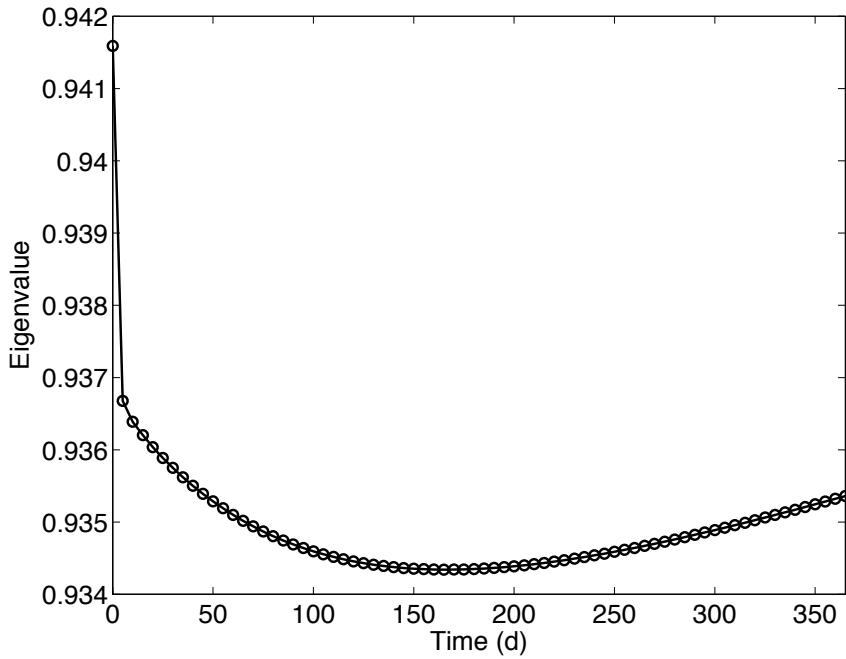


Figure 6.17: Eigenvalue over first year of NSC depletion calculation

Figure 6.18 illustrates the two-dimensional distribution of the power density at the beginning of the calculation. As might be expected, the center pin is producing most of the power, and the highest power density is concentrated on the edges of the center pin that are closest to the larger volumes of coolant (on the “corners” of the pin cell). This is where the thermal flux magnitude is largest as well. Figure 6.19 shows the power density profile across the center pin at both $t = 0$ and $t = 1$ year. The profile changes slightly in the direction of a flatter profile.

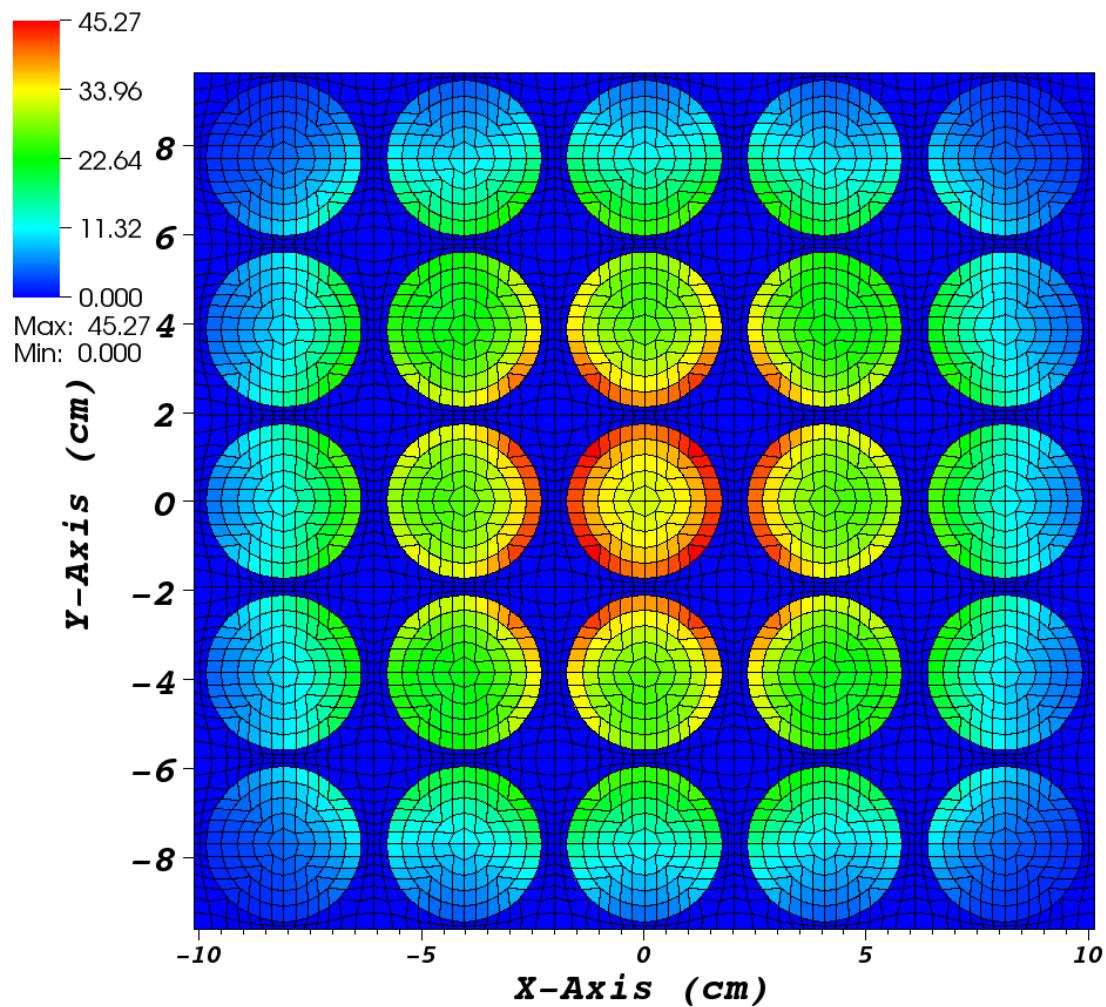


Figure 6.18: Power density profile at beginning of NSC depletion calculation (units: Watts/cc)

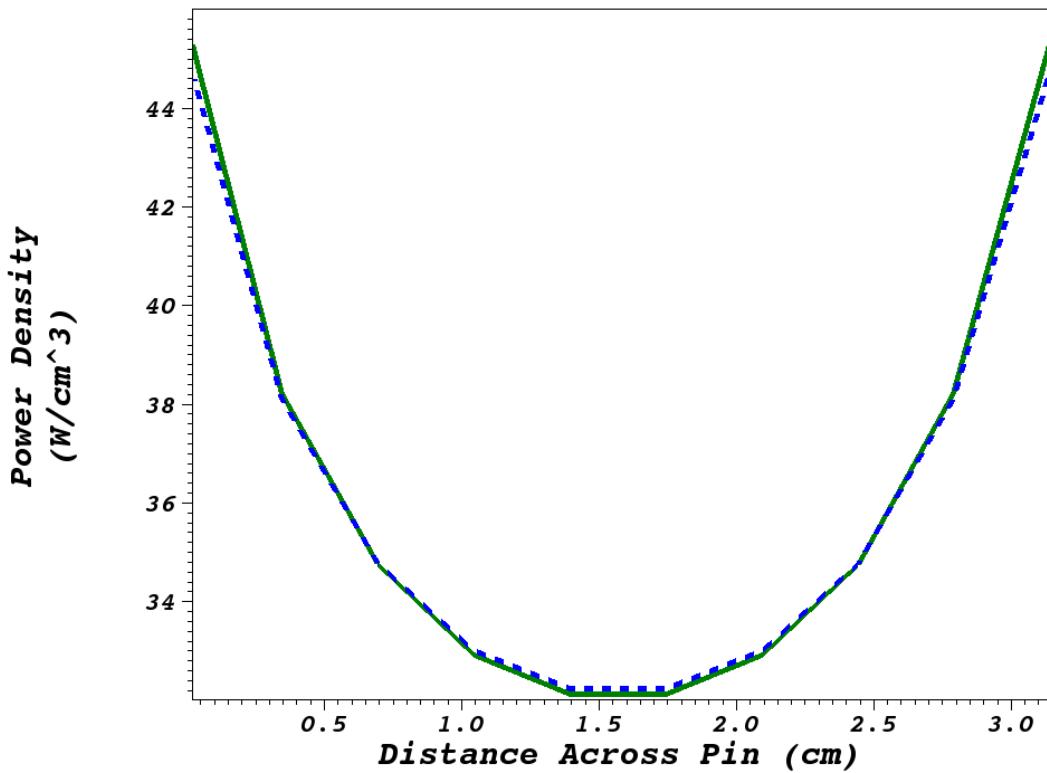


Figure 6.19: Comparison of NSC power density across center pin at $t=0$ (green, solid) and $t=1$ year (blue, dashed)

Figure 6.20 gives the Uranium-235 concentration over the center pin cell. We see that approximately 4-5% of the initial nuclide concentration is burned out in the center pin over the course of one year. In accordance with the power density profile, the largest depletion occurs at the edge of the pin, which is the reason that the power profile tends to flatten out. The jagged nature of the curves is a result of the averaging of the nuclide densities in each spatial cel.

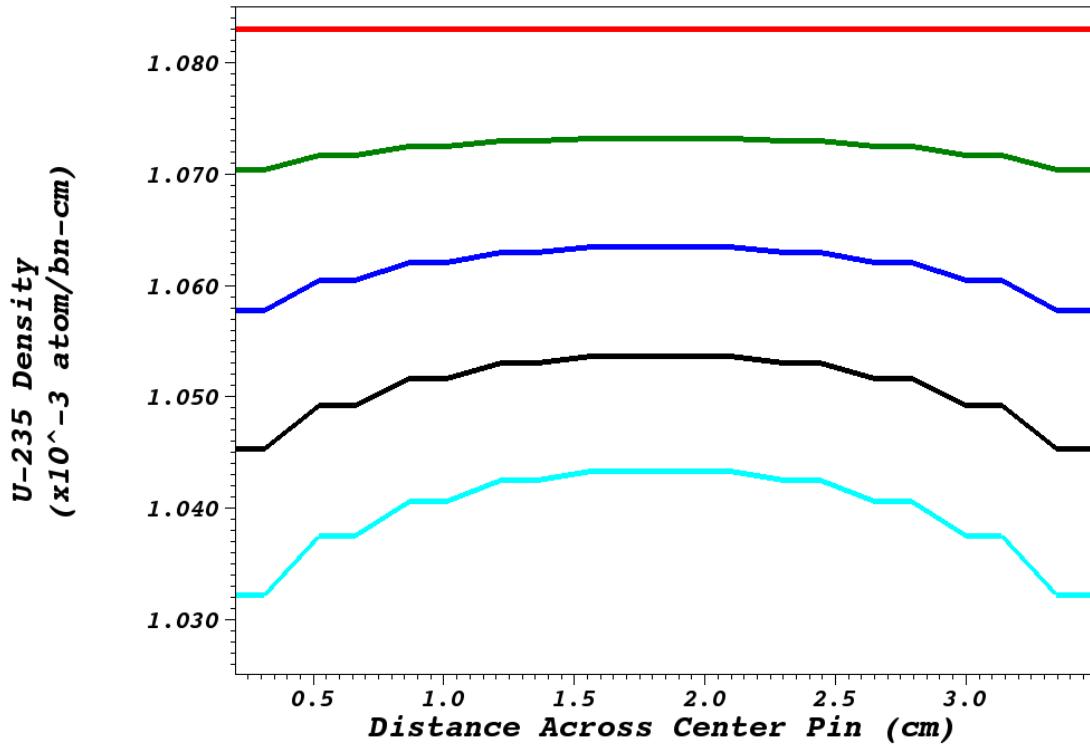


Figure 6.20: Uranium-235 number density over 1 year of NSC depletion. Line-out taken across center pin cell. Legend: $t=0$ (red), $t=0.25y$ (green), $t=0.5y$ (blue), $t=0.75y$ (black), $t=1y$ (cyan)

Figure 6.21 gives the Plutonium-239 concentration at 5 different time-steps during the simulation. The data for each line is taken by sampling a line at $y = 0$ with endpoints at $x = -2 \times \text{xPitch}$ and $x = 2 \times \text{xPitch}$. In this simulation, ^{239}Pu is formed after an $(\text{n},2\text{n})$ reaction in ^{238}U , which we model as producing ^{239}Np directly, and the subsequent decay of ^{239}Np . We note that profile of the burn-in of the plutonium roughly follows the expected profile of the neutron flux (and therefore $(\text{n},2\text{n})$ reaction rate). We expect that after longer depletion times, a large portion of the fissions will occur in ^{239}Pu . From this plot, it is not clear whether we are approaching an equilibrium or maximum ^{239}Pu concentration.

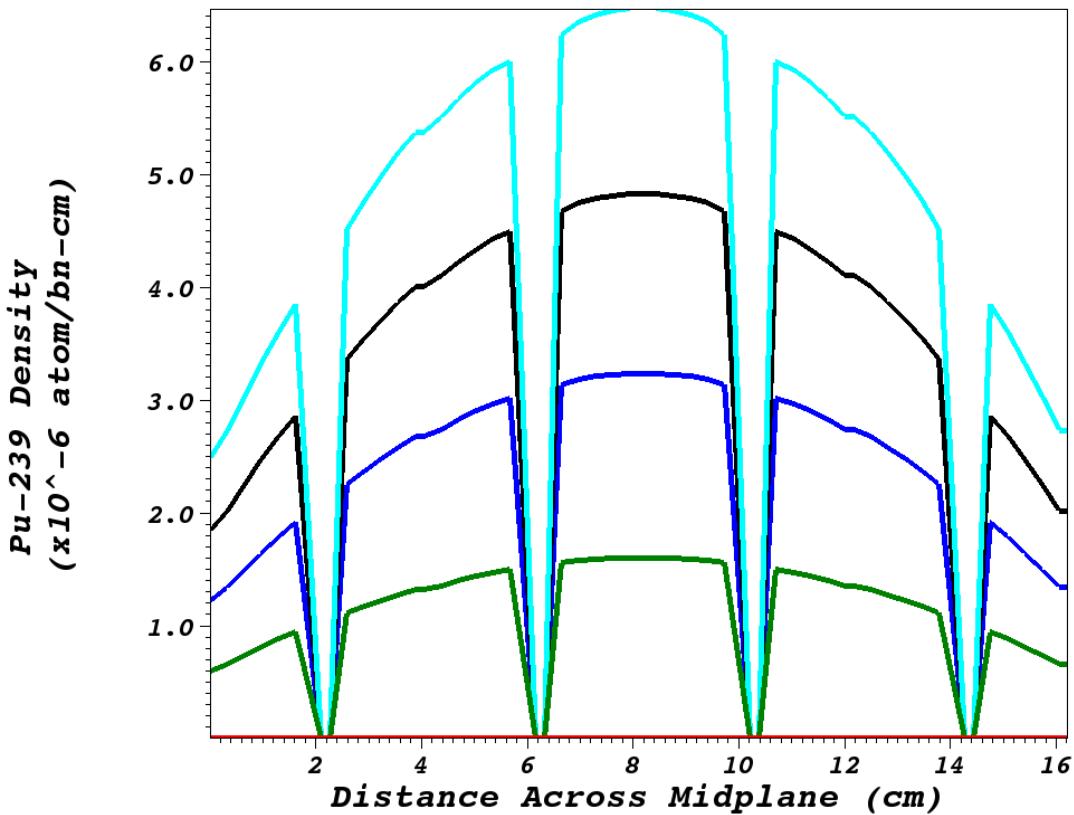


Figure 6.21: Plutonium-239 number density over 1 year of NSC depletion. Line-out taken from (x,y) points (-2,0) to (2,0) in units of x-pitch. Legend: t=0(red), t=0.25y(green), t=0.5y(blue), t=0.75y(black), t=1y(cyan).

Figure 6.22 gives the Samarium-149 concentration over the first year of depletion. The line-outs are taken from the same data as those in Fig. 6.21. Although we have not reached an equilibrium or maximum ^{149}Sm concentration, we appear to be close to one. An interesting feature of these profiles is that at early times, the maximum ^{149}Sm concentration within a pin is near the edges, where the fission rate is largest. After some time, however, the Samarium in these regions starts to burn off faster than the Samarium in the center of the pin, and the profile eventually flattens out.

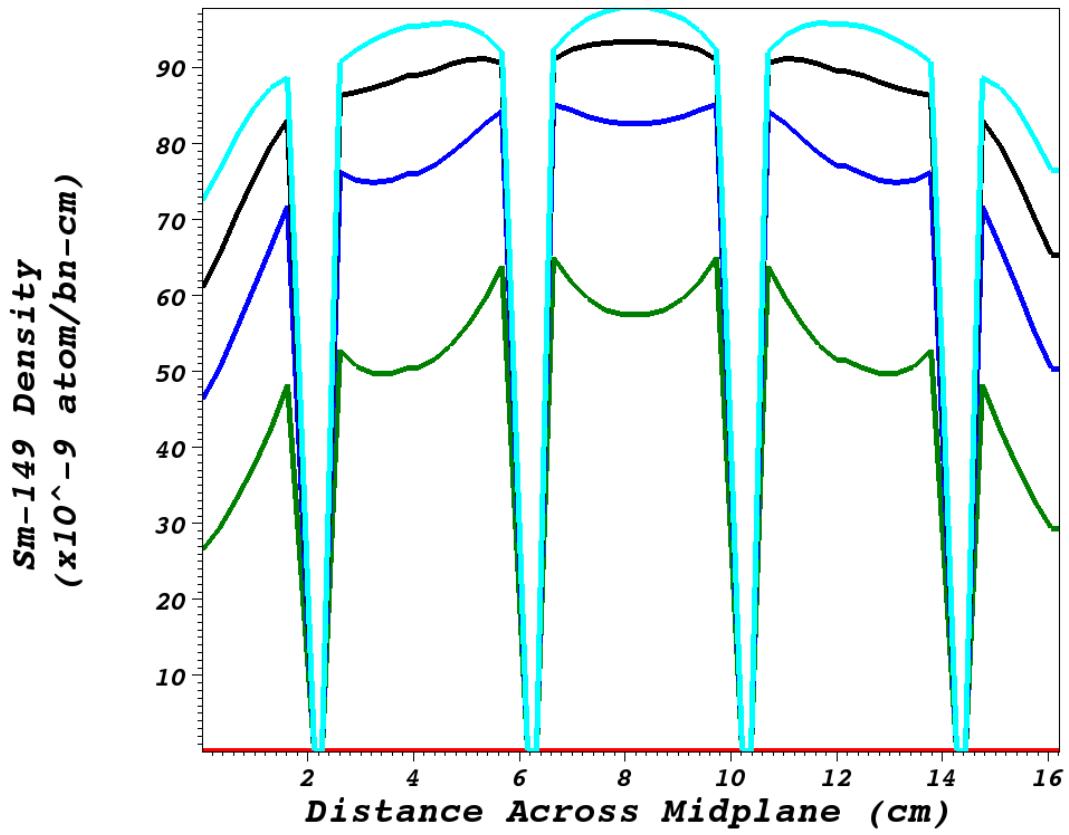


Figure 6.22: Samarium-149 number density over 1 year of NSC depletion. Line-out taken from (x,y) points (-2,0) to (2,0) in units of x-pitch. Legend: t=0(red), t=0.25y(green), t=0.5y(blue), t=0.75y(black), t=1y(cyan).

Finally, Fig. 6.23 shows the concentration of ^{135}Xe after 5 days of depletion. This profile does not change by more than 1% over the rest of the year of depletion; in other words, this is the computed equilibrium concentration. Although they have similar fission yields, ^{135}Xe reaches an equilibrium concentration faster than ^{149}Sm because of its short half-life.

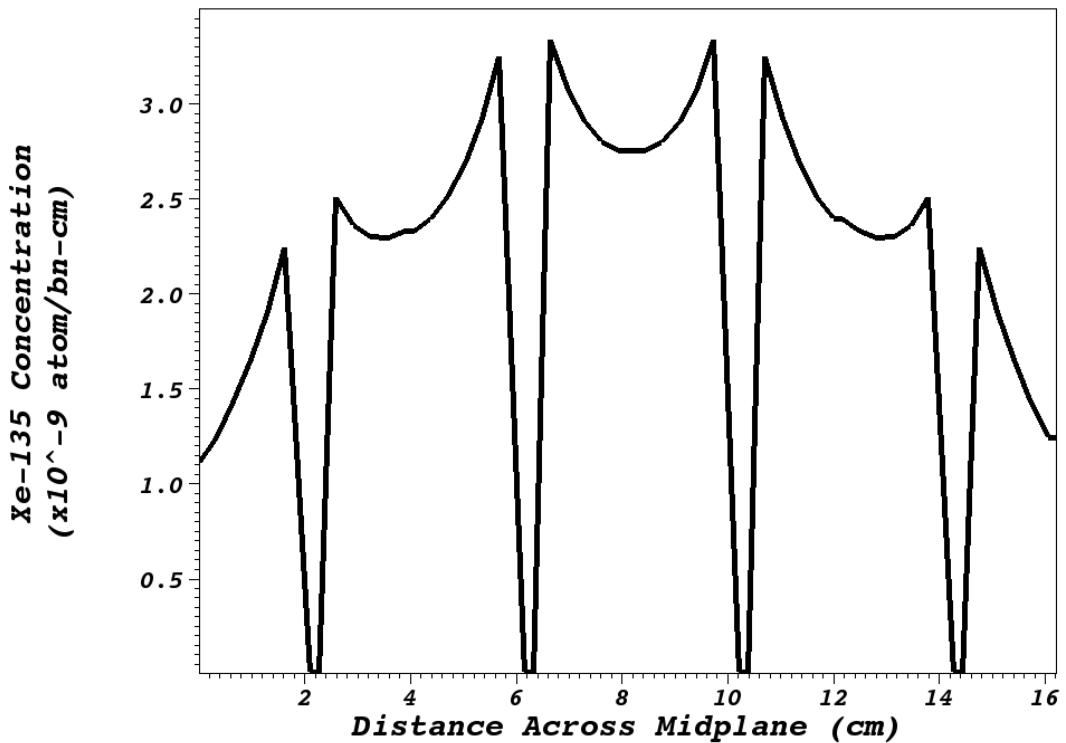


Figure 6.23: Xenon-135 steady-state concentration after 1 year of NSC depletion

A natural question is whether or not we have resolved the initial burn-in of Xenon well enough to predict the equilibrium concentration accurately. Although we are using 6 hour Bateman sub-cycles (the half life of Xenon is 9.2 hours), we are only updating the flux profile every 5 days. If the change in Xenon concentration that we see here affects the flux distribution and magnitude (it certainly affects the eigenvalue!), then we may be under-resolving the initial burn-in.

To test our resolution, the first 5 days of the simulation were re-run using 1 hour Bateman sub-cycles and half-day flux updates. Figure 6.24 compares the computed Xenon concentration at $t = 5$ days using the long and short time steps. The difference between the predictions is very small. Likewise, Fig. 6.25 compares the eigenvalue

of the system over the first 5 days using the two different time steps. The 5-day time-step model predicts $k_{\text{eff}}(5d)=0.9367$ while the half-day time step model predicts $k_{\text{eff}}(5d)=0.9365$, a relative difference of 2.1e-04, smaller than the eigenvalue tolerance of these runs.

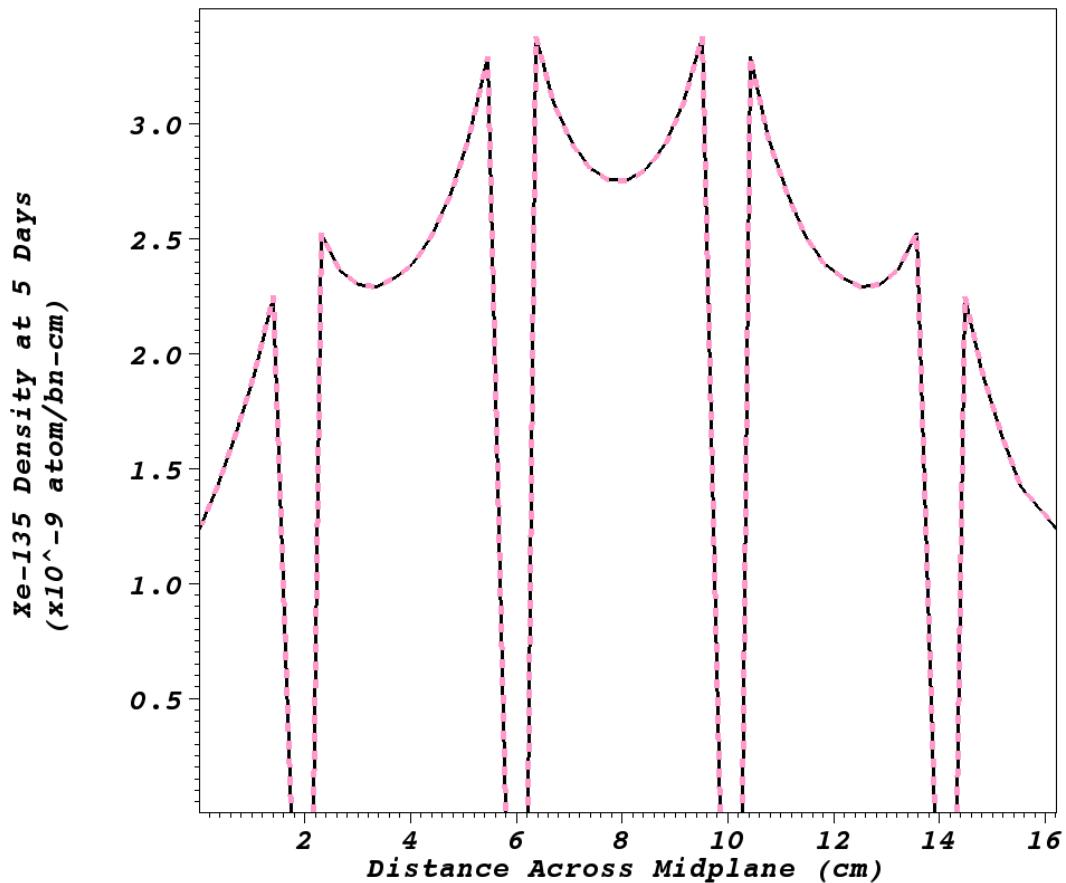


Figure 6.24: Comparison of ^{135}Xe concentration at 5 days using half-day and 5-day time steps. Line-out taken from (x,y) points (-2,0) to (2,0) in units of x-pitch. Legend: half-day(pink, dashed), 5-day(black, solid)

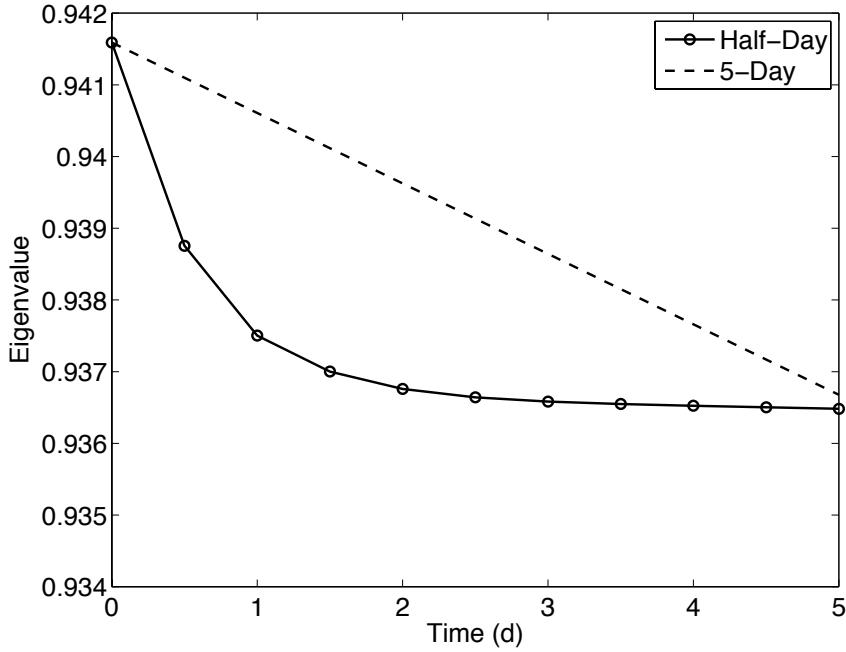


Figure 6.25: Comparison of k_{eff} at 5 days using half-day and 5-day time steps.

This problem demonstrates that the PDT depletion machinery developed as a part of this dissertation may be readily applied to the NSC simulations. We computed reasonable results using a moderate number of nuclides and one year of depletion time in less than 12 hours on a single cpu. The implicit nature of the time stepping scheme in combination with the ability to sub-cycle the Bateman solution allows for efficient times to solution. More detailed, 3D models of the NSC core will be available soon. Given appropriate choices for nuclides to track, power levels, and rod positions, the PDT depletion solver should be a valuable tool in recreating the operating history of the NSC. In the next subsection, we demonstrate how the adjoint capability may be used to further calibrate and reduce uncertainties in the model.

6.2.3 NSC depletion-perturbation calculations

After a suitable depletion model is reached, we may wish to use the depletion-perturbation methods outlined in this dissertation to compute the sensitivity of the QOIs with respect to initial densities, nuclide cross sections, and other parameters in the calculation. In this subsection, we describe an example depletion-perturbation problem and compare the computed sensitivities with those obtained by a brute-force sampling/finite difference method.

The example problem has a 2 by 2 pin layout. The fuel material has nominal initial densities given in Table 6.8; the moderator material has ^1H and ^{16}O densities as listed in Table 6.9 with the addition of ^{10}B with density 1.0e-5 [$\frac{\text{atom}}{\text{b}-\text{cm}}$]. The power level was held constant at 9kW, and the system was depleted for 36 time-steps of 5 days each using 1/4 day Bateman sub-cycles and the implicit Euler time stepping scheme. The QOI was the terminal reactivity, $\rho(t_f) = \frac{k_{\text{eff}}(t_f)-1}{k_{\text{eff}}(t_0)}$.

The sensitivity of this QOI was computed with respect to a wide range of parameters. One parameter was the initial density of each nuclide present in the fuel and water. Figure 6.26 shows how the trajectory of k_{eff} changes due to a 5% increase in each of 7 nuclides in the problem. As expected, the reactivity is most sensitive to the primary fuel and moderator densities. The true and adjoint-predicted changes in the reactivity as due to these perturbations and the relative difference between them are presented in Table 6.12.

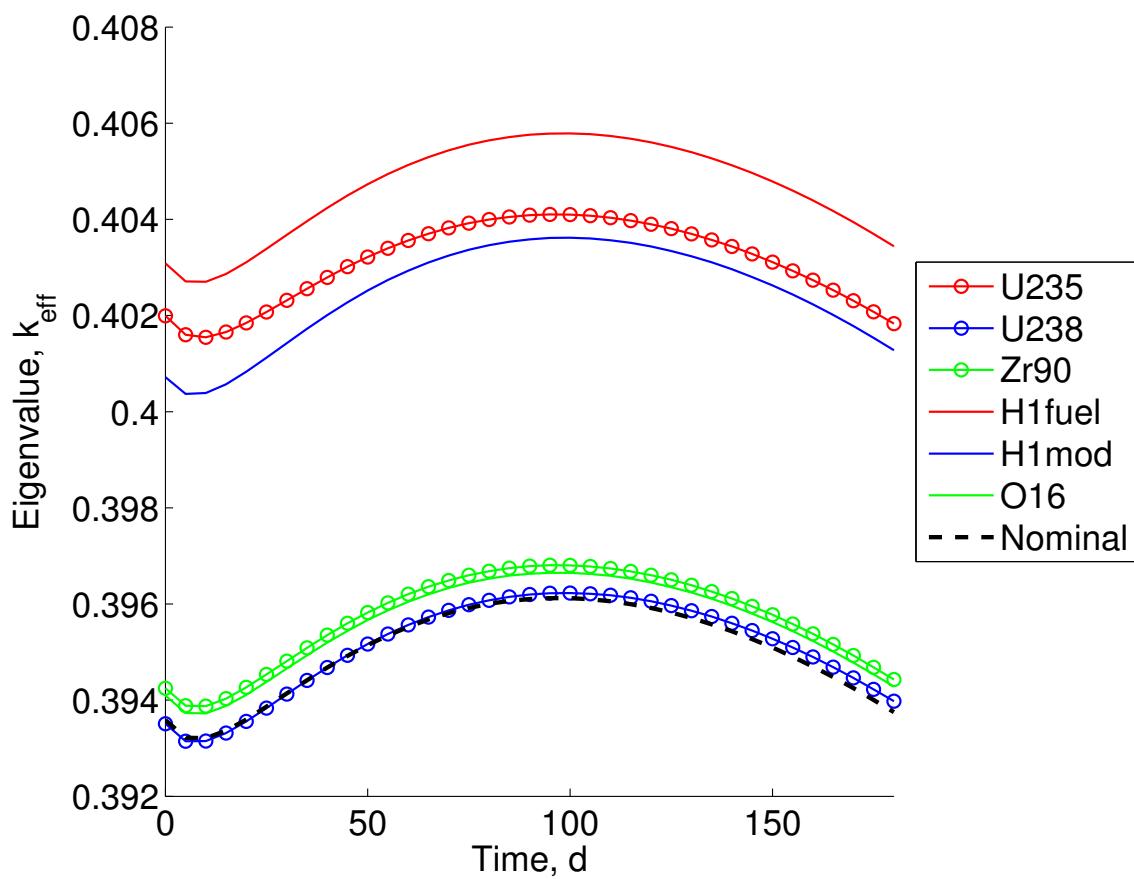


Figure 6.26: Sensitivity of final reactivity with respect to nuclide initial density.

Table 6.12: Finite difference and adjoint-based estimates for reactivity jumps due to 5% initial condition perturbations

Parameter	True Jump	Predicted Jump	Relative Diff
^{235}U	5.107e-02	5.341e-02	4.5%
^{238}U	1.463e-02	1.217e-03	-16.8%
^{90}Zr	4.361e-03	4.383e-03	0.5%
$^1\text{H} - \text{fuel}$	6.100e-02	6.383e-02	4.6%
$^1\text{H} - \text{mod.}$	4.765e-02	5.004e-02	5.0%
^{16}O	3.425e-03	3.452e-03	0.8%

We also computed the sensitivity of this QOI with respect to a wide range of cross sections. If we make a simple assumption that all cross sections are known to within a certain percentage, then sensitivity the measure

$$S = p \frac{d\rho}{dp}$$

can be used to rank all parameters in terms of their importance in predicting the QOI. We found for this problem that the cross sections for ^{235}U and ^1H consistently had the highest sensitivity measures, and a subset of these parameters are ranked according to the absolute value of this metric in Table 6.13. The (n,2n) cross section for ^{238}U and the fission and neutrons-per-fission data for ^{239}Pu also had notably high values of S and would likely play a more important role in longer depletion calculations. These results agree with intuition. It is the fission and slowing-down phenomena that determine reactivity in a thermal spectrum reactor; also, as initial ^{235}U material burns away and ^{239}Pu material is generated, the relative amount of

fission in the plutonium nuclides increases.

Table 6.13: Relative sensitivity of nuclear data in NSC simulation

Component	Parameter	Sensitivity Measure, S
^1H	$\sigma_{s,2 \rightarrow 2}$	6.01
^1H	$\sigma_{t,2}$	-5.96
^1H	$\sigma_{t,1}$	-5.84
^1H	$\sigma_{s,1 \rightarrow 1}$	5.82
^{235}U	ν_2	1.89
^{235}U	$\sigma_{f,2}$	1.85
^{235}U	$\sigma_{t,2}$	-1.45
^1H	$\sigma_{s,1 \rightarrow 2}$	1.18

This application problem served two purposes. First, we demonstrated that the depletion perturbation machinery works on the NSC problem and generates results that agree with direct finite difference parameter perturbations. The second purpose was to give an idea for applying the computed sensitivities to determine which dimensions in the stochastic space are most responsible for variation in the output. These are dimensions that may require further investigation, such as more dense sampling, in a UQ study. Future depletion perturbation simulations of the NSC core can use this investigation as a starting point.

6.3 Checkpointing Scheme Scaling Study

In this section, we describe a study of the performance of five checkpointing schemes as both the number of processors and the number of unknowns per processor are increased towards the levels required for high-fidelity reactor analysis calculations. We described the five schemes in Sec. 4.6, including the three new schemes that checkpoint and store only converged source moments. We also presented models for the computational cost and RAM footprint of each scheme. Here, we test our performance models and compare the time to solution and scalability of the schemes for a particular source-driven depletion perturbation problem.

6.3.1 Description of the source-driven test problem

We test the schemes using a source-driven depletion perturbation problem in order to compare the experimental results to the performance models in Sec. 4.6, which were written for the source-driven case. The test problem has 3D brick-cell geometry with total x, y, and z dimensions of 30.0cm. 30.0cm, and 10.0cm, respectively. The material is a homogenous mixture of the seven nuclides listed along with their initial number densities in Table 6.14. The depletion physics are meant to emulate ^{239}Pu production, where we again assume that an (n,γ) reaction in ^{238}U produces ^{239}Np directly. The problem is run for 6 time-steps of approximately 14 hours using the explicit-Euler Runge-Kutta scheme. There are 11 energy groups, and the fixed isotropic source has intensity $1.8\text{e}+12 \left[\frac{\text{n}}{\text{cm}^3\text{-sec}} \right]$ in each group. We use S_4 level-symmetric quadrature to discretize the angular variable (24 total discrete ordinates).

Table 6.14: BOL nuclide list and densities for checkpointing scheme scaling problem
 (density units: $\left[\frac{\text{atom}}{\text{b}-\text{cm}}\right]$)

Nuclide	Density
^{235}U	6.5e-03
^{238}U	1.6e-01
^{239}Np	1.5e-01
^{239}Pu	3.0e-02
MU-35	3.6e+00
MU-38	2.9e+00
MPu39	2.9e+00

The 11-group cross sections for the scaling problem were collapsed from the 33-group cross sections produced by Argonne National Laboratory for the traveling wave reactor benchmark problem. To collapse the 33-group data, we used the EOEC flux spectrum taken from the center cell of the traveling wave reactor problem. Using the 11-group cross sections and the initial densities given in Table 6.14, we find that 10cm is approximately 800 mean-free-paths in the fastest group and 1500 mean-free-paths in the slowest group and that the optical thickness in the other groups varies between these depths. The BOL k-eigenvalue of the system is approximately 0.86. We consider only isotropic scattering.

We perform weak scaling by increasing the number of spatial cells owned by each processor in a particular problem. We consider each combination of three processor counts and three problem sizes. The three processor counts are 1024 (1k), 2048 (2k), and 4096 (4k). The three problem sizes result in approximately 200,000 (200k), 400,000 (400k) and 800,000 (800k) angular flux unknowns per processor. The

break-down of the 9 different problems is given in Table 6.15 in terms of number of processors (P) and number of cells in x, y, and z (N_x , N_y , N_z), total number of angular flux unknowns per cpu, N_ψ/cpu , total number of unknowns per cpu, $N_{\text{tot}}/\text{cpu}$, and the RAM footprints of a single copy of the angular flux, M_ψ , and single copy of the source moments, M_S , per processor.

Table 6.15: Variation in unknowns per processor for scaling study. Note: each problem has 7 nuclide densities per cell, 8 spatial unknowns per cell, 24 angles, and 11 energy groups.

Problem	P	N_x	N_y	N_z	N_ψ/cpu	$N_{\text{tot}}/\text{cpu}$	M_ψ (MB)	M_S (MB)
1	1024	32	64	48	202,752	203,424	1.62	0.068
2	1024	64	64	48	405,504	406,848	3.24	0.135
3	1024	64	64	96	811,008	813,696	6.48	0.270
4	2048	64	64	48	202,752	203,424	1.62	0.068
5	2048	64	64	96	405,504	406,848	3.24	0.135
6	2048	64	128	96	811,008	813,696	6.48	0.270
7	4096	64	64	96	202,752	203,424	1.62	0.068
8	4096	64	128	96	405,504	406,848	3.24	0.135
9	4096	128	128	96	811,008	813,696	6.48	0.270

We ran 11 repetitions of each of the nine problems, where each repetition used a different checkpointing scheme. The 11 schemes were STOR_ALL, STOR_MOM, CKPT_ALL_K, CKPT_MOM_K, and INTP_MOM_K, with K=2, 3, and 4. With 6 total time steps, the K=2, 3, and 4 schemes will write a total of 4, 3, and 3 checkpoint files, respectively. The K=2 schemes will write a file at $t = t_0$, then after time steps

2, 4, and 6. The K=3 schemes will write a file at $t = t_0$ and after time steps 3 and 6. The K=4 scheme will write a file at $t = t_0$ and after time step 2. In this case, the last four time-steps will be stored during forward-mode, which is the most optimal configuration in terms of minimizing re-compute costs.

Although we are not particularly interested in the solution to these problems, we do ensure that the schemes produce consistent results. For each problem, we compute the sensitivity of a total inventory QOI with respect to all the microscopic total cross sections in the problem. The sensitivities for the STOR_ALL, STOR_MOM, CKPT_ALL, and CKPT_MOM schemes always agree to machine precision. The relative difference between these schemes and the INTP_MOM schemes is not machine precision because of the interpolated initial guess for the source moments during adjoint mode; instead, the sensitivities for the STOR_ALL scheme and INTP_MOM scheme always agree to at least solver tolerance.

6.3.2 Scaling results

The 11 repetitions of each problem were run on the vulcan machine at Lawrence Livermore National Laboratory. This machine has 16 cpus/node and 16GB of RAM/node (1 GB/cpu). The parallelism was always such that 16 MPI processes were launched per node. We allowed PDT's auto aggregation and partitioning algorithm to choose the partitioning parameters to optimize time to solution for each problem. Machine queue limitations at the time of this study limited run-times to 12 hours; as a result, some of the 800k runs did not complete. We are still able to make conclusions about the schemes without these results.

Figure 6.27 shows the observed memory footprint for each of the 11 schemes and the 400k problem size (the plot is the same for each cpu count). Each footprint is in agreement with the corresponding prediction given in Table 4.2 (plus a small

allocation for the densities). The parameters S and T for this problem are each 1, and the parameter N is 6. For example, the STOR_ALL scheme requires a total of seven allocations for ψ and one allocation for ψ^\dagger , or $8 \times 3.24 = 25.92$ MB. Likewise, the STOR_MOM scheme requires one allocation for ψ , one for ψ^\dagger , and seven allocations for the source moments; that is $2 \times 3.24 + 7 \times 0.135 = 7.4$ MB. The CKPT_ALL_2 scheme requires a total of 5 allocations of size ψ , yielding the allocation of 16.3MB of storage. The other schemes also agree with the predictions.

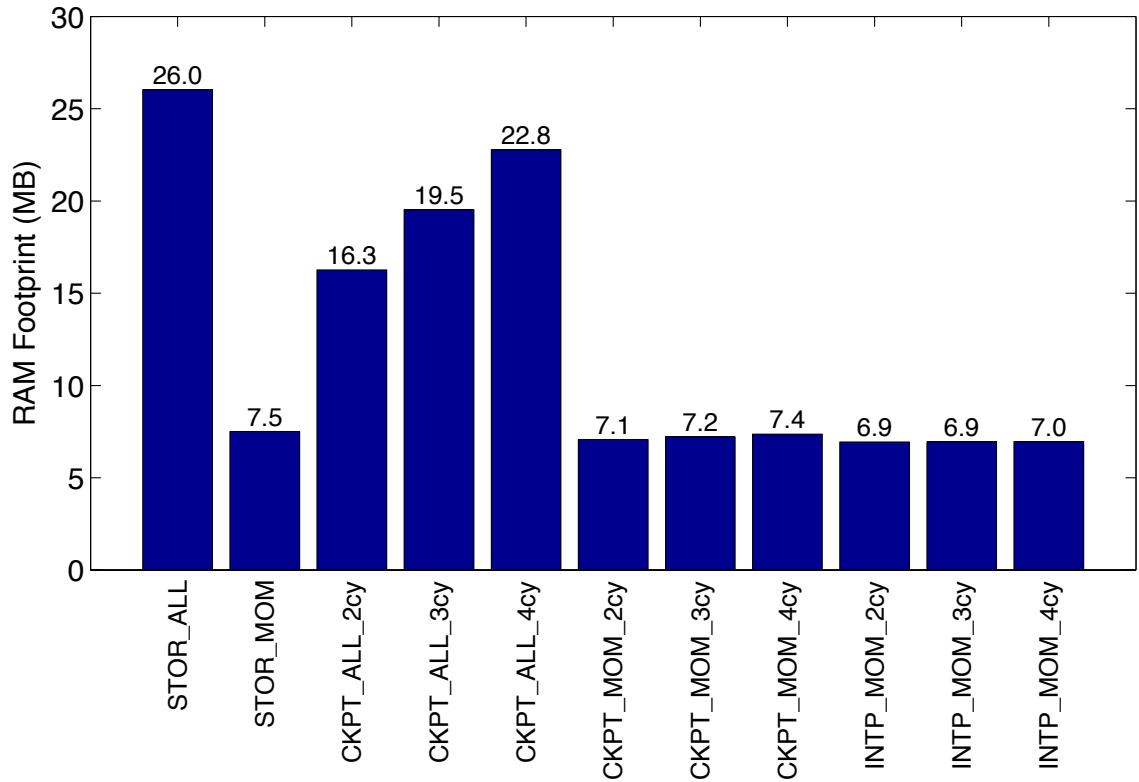


Figure 6.27: Scheme memory footprint for the 400k unk/cpu problem.

We draw several broad conclusions from Fig. 6.27. First, the STOR_ALL scheme

is useful only as an academic exercise. Real problems will require hundreds of time steps with more unknowns per cpu and thus will quickly exceed the machine's RAM capacity. The STOR_MOM scheme is viable to a degree. As implemented, it will always incur the $2M_\psi$ cost, but if that fits and the degree of anisotropy is fairly low, the source moments may fit in RAM for smaller or modest-sized problems. The relative difference between the RAM footprints of the STOR_ALL and STOR_MOM schemes will grow linearly with N for a given problem. In other words, if our test problem had more time steps, the STOR_MOM scheme would look even more attractive compared to the STOR_ALL scheme in terms of memory footprint.

Other conclusions are related to the growth of the memory footprint as K grows. The memory footprints of the CKPT_ALL and CKPT_MOM schemes grow linearly with K , but the slope of that growth is M_ψ for the CKPT_ALL scheme while only M_S for the CKPT_MOM scheme. For a uniform quadrature set (that is, same number of angles in each energy group), the ratio $M_\psi : M_S$ is equal to the ratio of the number of discrete ordinates to the number of angular moments. For this problem, that ratio is 24, yielding the much faster growth rate for the CKPT_ALL scheme. For other problems, the ratio may not be as high, but will always be greater than 1. The INPT_MOM scheme, on the other hand, does grow linearly with K , but only with a slope equal to the memory footprint of the Bateman solution, which is typically much smaller than the flux solution. That growth is barely detectable in the figure.

Next we verify that the number of fixed-source solves required for the forward, adjoint, and recompute modes is as prescribed by Table 4.1. Figure 6.28 shows the fixed-source solve count for each mode for the 200k/2k problem (the count is the same for every problem size on every processor count). Recall that the analysis in Sec. 4.6 assumed that $N_R = N/K$ is an integer, which is not the case in this problem for $K = 4$. The analysis does hold, however, for $K = 2$ ($N_R = 3$) and $K = 3$

$(N_R = 2)$.

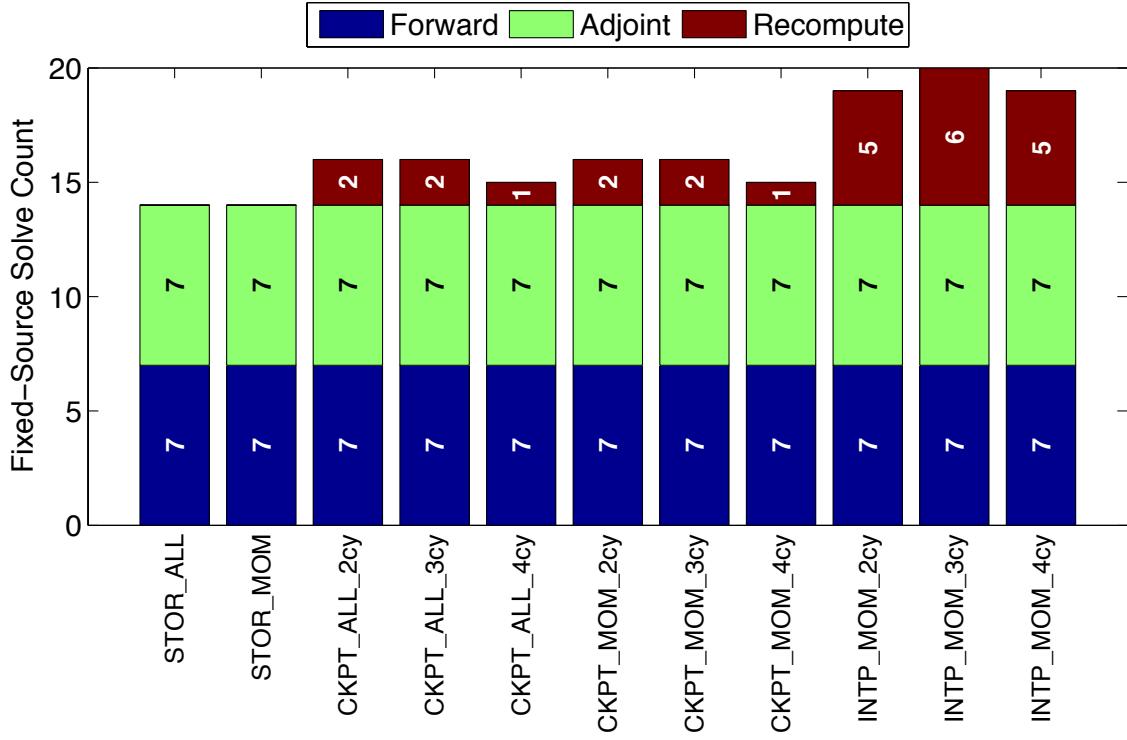


Figure 6.28: Fixed-source solve count for the checkpoint scheme scaling study

First, each scheme requires exactly seven forward mode and adjoint mode fixed-source solves, as predicted. The computational work varies from scheme to scheme only in the recompute cost. The STOR_ALL and STOR_MOM schemes require zero recompute fixed-source solves. As predicted by Table 4.1, the CKPT_ALL and CKPT_MOM schemes require two recompute fixed-source solves for $K = 2$ and $K = 3$. For $K = 2$, the fluxes that are recomputed are the end-of-first and end-of-third time step fluxes. The $K = 3$, the recomputed fluxes are located at the end of the first and second time steps. For $K = 4$, only the end-of-first flux is recomputed. A file is written after the second time step, and the end-of-third, fourth, and fifth

time step fluxes are stored during forward mode.

The recompute fixed-source solve counts for the INTP_MOM schemes are also as predicted by the analysis. For example, the five recompute solves for INTP_MOM_2 are (in sequential order) end-of-fifth time step (interpolated), end-of-third time step, end-of-third time step (interpolated), end of first time step, end of first time step (interpolated).

We are interested to compare the time to solution for the CKPT_ALL_2 and CKPT_ALL_3 schemes (and likewise the CKPT_MOM_2 and CKPT_MOM_3 schemes). These schemes have different memory footprints but the same total number of recompute fixed-source solves (they just occur at different time steps). Time to solution will be discussed below.

Figures 6.29 and 6.30 show the total number of sweeps and sweeps per fixed-source solve for each mode and each scheme for the 400k/2k problem. First, we note that the total number of sweeps (and therefore total number of sweeps per fixed-source solve) in forward and adjoint modes is the same for each scheme. This is in some ways a strong verification result because it indicates that each of the checkpointing schemes is delivering the same information to the adjoint fixed-source solver. If, for example, differences in the schemes resulted in a slightly different adjoint transport source, it is possible that the total number of sweeps to converge the flux solution would differ. We do not see evidence of this kind of bug here.

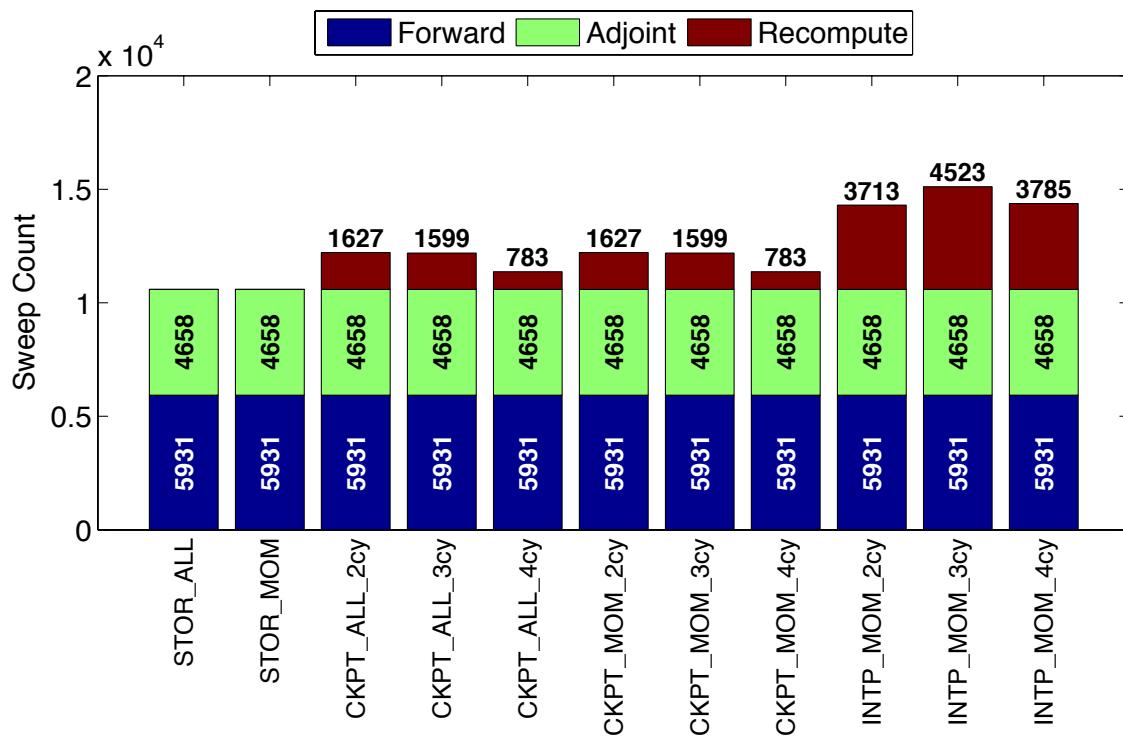


Figure 6.29: Sweep count for the checkpoint scheme scaling study

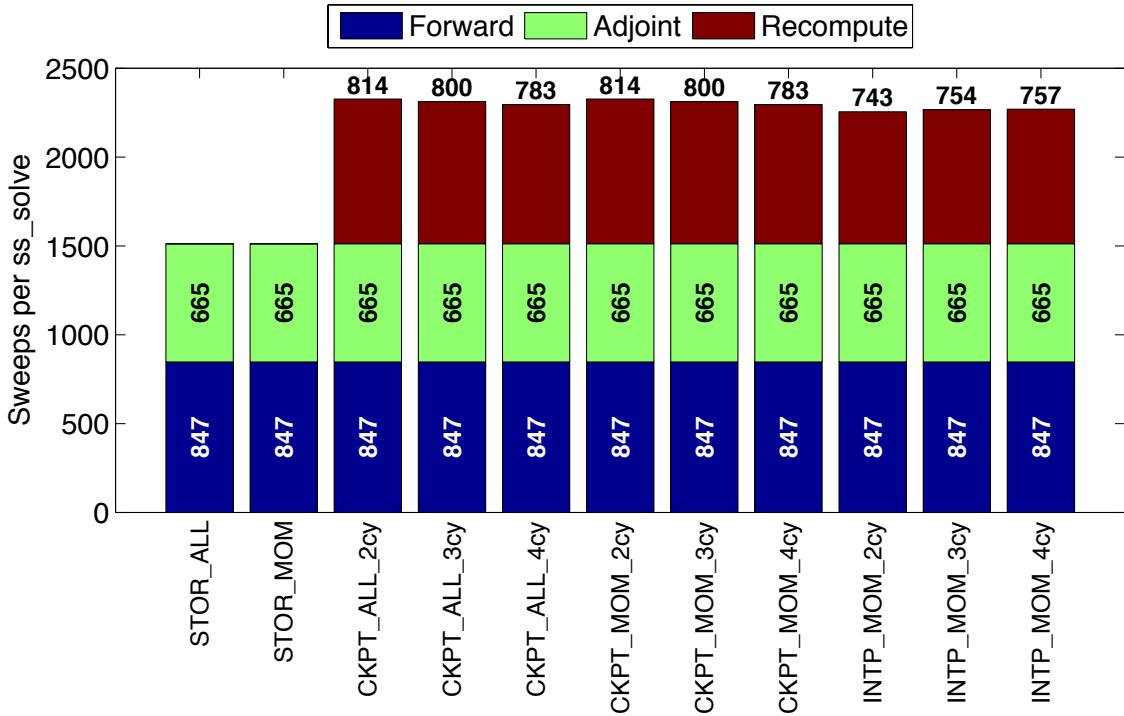


Figure 6.30: Sweeps per fixed-source solve for the checkpoint scheme scaling study

Another verification note is that the CKPT_ALL and CKPT_MOM schemes are performing the exact same number of sweeps to recompute the forward solution. They should be: the only difference between these schemes is the data that is written to and from file and is stored in RAM; their methods to kick-off a recompute segment are identical. The variation in the number of sweeps per solve as K changes can be explained by the contiguity of the time-steps which are re-computed. For $K = 3$, the end-of-first and end-of-second time steps are recomputed sequentially, so the initial guess for the end-of-second flux is well informed. For $K = 2$, the end-of-first and end-of-third fluxes are recomputed, but not sequentially and without any carryover of the initial guess. For $K = 4$, only the end-of-first flux is recomputed, and it is not clear why it requires less sweeps to converge.

A final important note is that the INTP_MOM schemes on average requires fewer

sweeps per recompute fixed-source solve. This is encouraging because it indicates that our scheme for interpolating the source moments to form an initial guess for the total source is saving some computational effort. Although the absolute number of sweeps is on the order of twice as much, each fixed-source solve is slightly less expensive.

Figure 6.31 shows the number of forward-mode and adjoint-mode single recovery sweeps required by each scheme to solve the problem. The schemes that store or checkpoint the full ψ solution do not require any single sweeps. The STOR_MOM scheme is straightforward to understand: a single forward-mode recovery sweep is required at each time step (except the last) to reproduce and store ψ . The CKPT_MOM and INTP_MOM schemes follow the predictions made by Table 4.1. For example, in the CKPT_MOM_2 scheme, 6 total forward sweeps are required during adjoint mode to recompute ψ at the particular time step. The other 4 sweeps are required during recompute mode to kick off and complete two of the recompute segments. The INTP_MOM schemes generally require less single sweeps because they do not store the source moments except for at the beginning and end of a recompute segment. Finally, as a reminder, the single adjoint sweeps are used to kick off adjoint mode after a recompute segment.

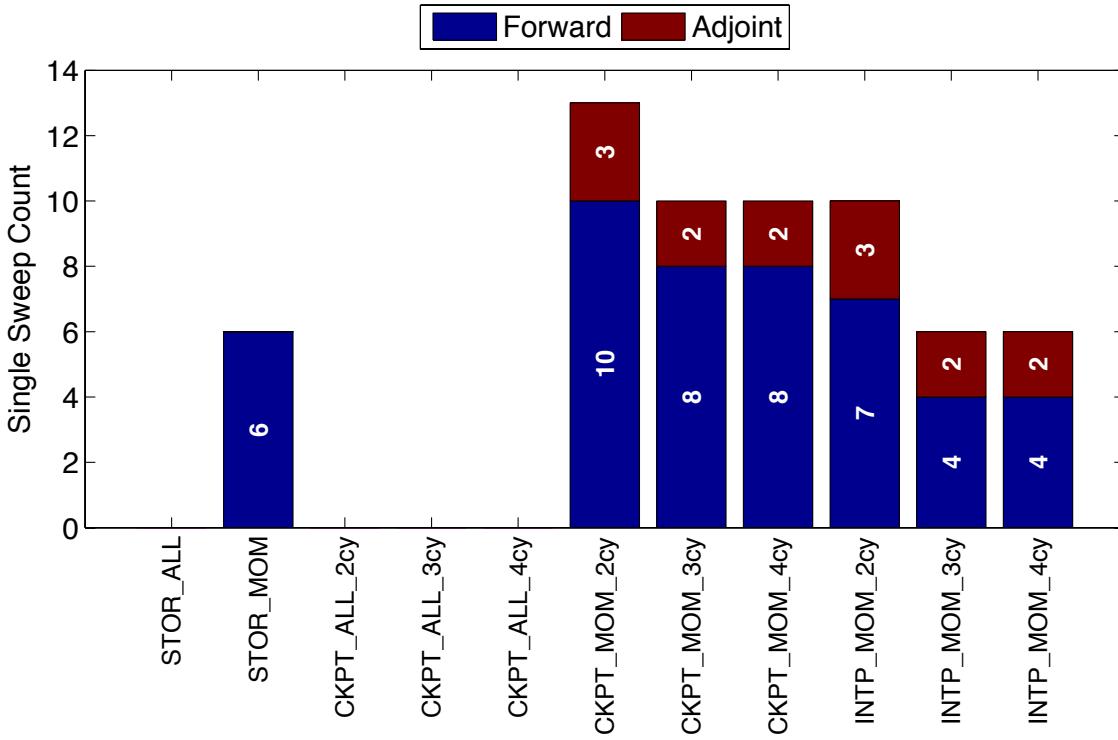


Figure 6.31: Recovery sweep count for the checkpoint scheme scaling study

We now turn to time to solution and scaling results. Some of the schemes for the 800k problem size did not complete because their run times exceeded the queue limitation on our host machine. The STOR_ALL and STOR_MOM schemes finished for all problem sizes and all processor counts; only a small number of the 800k problems with file checkpointing completed. The runs that did complete, however, provide us with enough information to characterize the relative performance of the schemes.

Figure 6.32 shows the time required to solve the 400k problem using 4k processors for each of the 11 repetitions. The relative difference between the different schemes for this problem size and processor count is representative of all 9 problems. The relative differences between the solution times correspond very closely to the relative

differences between the sweep counts (computational work) shown in Fig. 6.29. The STOR_ALL and STOR_MOM schemes require the least amount of work and have the shortest run times. The INTP_MOM schemes require the most re-calculation work and therefore take the longest to complete.

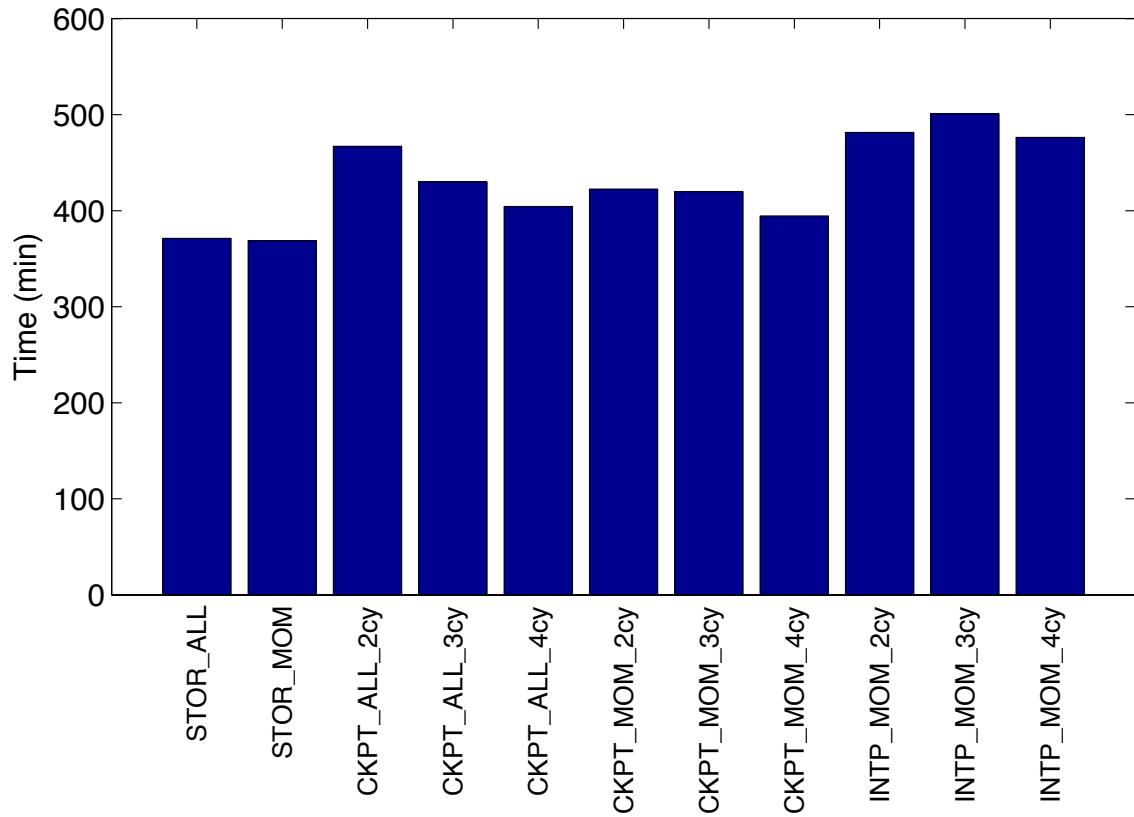


Figure 6.32: Time to solution for the checkpoint scheme scaling study (400k unknowns per processor, 4k processors)

There is one discrepancy in these timing results. The CKPT_ALL_2cy scheme requires somewhat longer time to solution than the computational work would suggest. We expect it to take roughly the same amount of time as the CKPT_ALL_3cy scheme, similar to the comparison between the CKPT_MOM_2cy and CKPT_MOM_3cy schemes.

This discrepancy appeared for all 9 problem configurations. Results to date indicate that the extra time for this scheme is spent in the calculation of $\left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - G\psi] \right\rangle_{E,D,\Omega}$ in the adjoint Bateman equation. We are working to confirm this and understand why the calculation is slower for this scheme.

Although work to diagnose the reason for this discrepancy is ongoing, its presence does not prevent us from drawing conclusions about the performance of the checkpointing schemes. We claim that the STOR_MOM, CKPT_MOM, and INTP_MOM schemes are favorable alternatives to the STOR_ALL and CKPT_ALL schemes. For this problem, the number of recovery sweeps is small compared to the total number of sweeps required to solve the problem; the memory footprint reduction, however, is significant for the new schemes. Therefore, the computation-for-memory tradeoff is certainly favorable. The MOM schemes will scale to larger problem sizes before running out of memory, and the extra work they incur does not result in significantly longer times to solution.

Next we look at the weak scaling performance of the different schemes. Documented PDT scaling results[1, 2] show efficiencies of greater than 90% when scaling from 8 to 256,000 processors using a test problem with about 2.5e+06 unknowns per processor. We wish to make sure that our weak scaling results are on par with this level of performance. Figure 6.33 shows weak scaling results using the 400k problem for the STOR_MOM and CKPT_MOM schemes. The schemes scale nearly identically for this problem and achieve greater than 90% weak scaling efficiency at 4000 processors compared to 1000 processors. We expect that this efficiency would increase to known PDT performance given a test problem with more unknowns per processor.

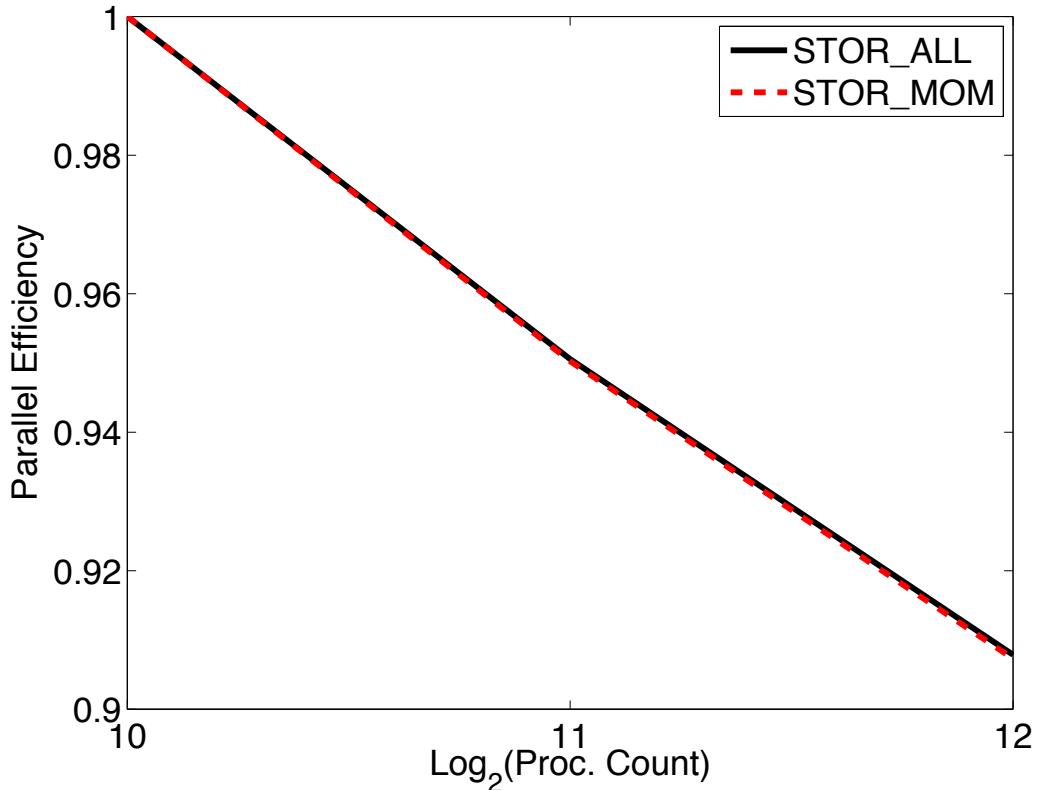


Figure 6.33: Weak scaling results for the STOR_ALL and STOR_MOM schemes (400k unknowns per cpu)

Figure 6.34 shows the parallel efficiency for the CKPT_ALL and CKPT_MOM schemes for the same 400k problem. The efficiencies for these schemes fall between 88% and 90% when moving from 1000 to 4000 processors (with the exception of CKPT_ALL_2, the results for which we believe to be inconsistent). These slightly lower efficiencies are likely a result of the file I/O required for these schemes, and we believe the algorithm for performing this I/O can be improved in future work. The results within a scheme are consistent: CKPT_ALL_4 is slightly more efficient than CKPT_ALL_3, and CKPT_MOM_4 is slightly more efficient than CKPT_MOM_3. This further suggests that I/O is causing the inefficiency because the CKPT_ALL_4

and CKPT_MOM_4 schemes write one less checkpoint file each than the CKPT_ALL_3 and CKPT_MOM_3 schemes. Finally (except again for the CKPT_ALL_2 scheme), the MOM schemes are slightly more efficient than the ALL schemes at 4000 processors, possibly because the MOM schemes read and write less from and to disk at each file I/O step.

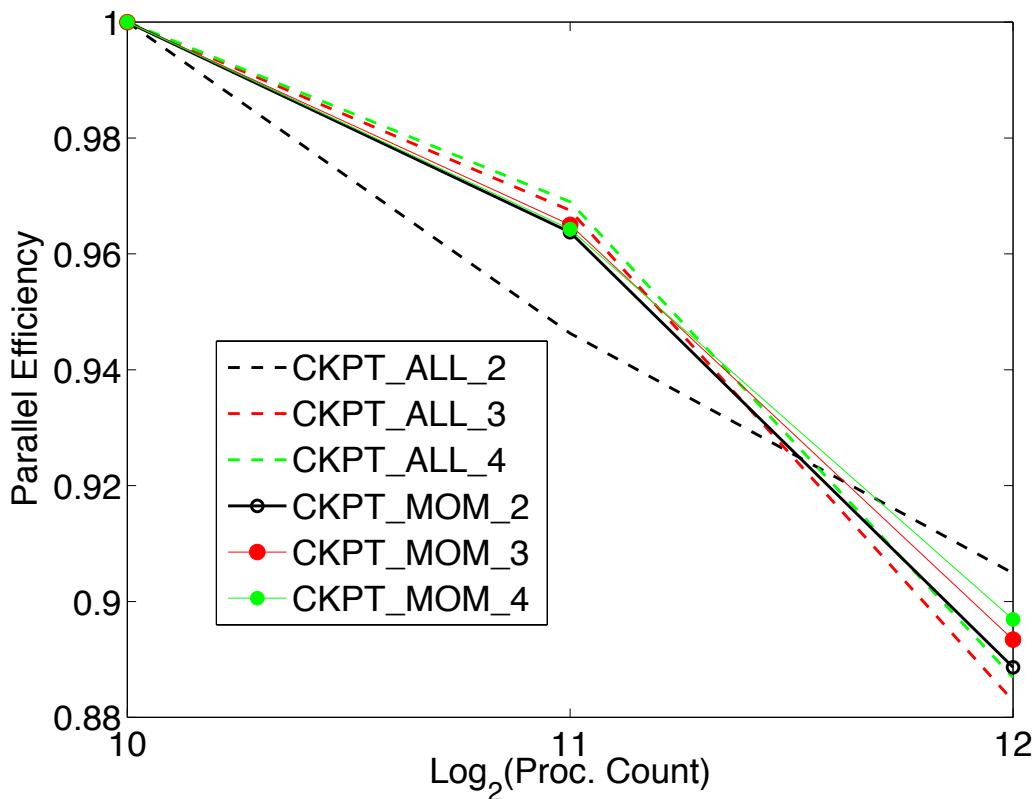


Figure 6.34: Weak scaling results for the CKPT_ALL and CKPT_MOM schemes (400k unknowns per cpu)

We are also interested to see how the time to solution is affected as we increase the number of unknowns but keep the number of processors the same. In the limit of a large number of unknowns, we would expect the performance of the STOR_ALL and

CKPT_ALL schemes to rapidly deteriorate because the forward solution allocations would swamp the memory capacity of the machine. We were unable to approach this limit in this study, unfortunately, but we give our preliminary results below.

Figure 6.35 shows the time to solution on 2k processors as the problem size is increased (only for those schemes that completed the 800k problem in the allotted time). In this efficiency measure, the schemes are indistinguishable. We do note that PDT becomes more efficient as the number of unknowns increases. This behavior, which is predictable using PDT's performance models, is related to decreased idle time of the processors because they have more work to do between communications with other processors. The increase in efficiency increases for problem sizes approaching the usual PDT scaling problem, which has about twice as many unknowns as the largest problem size shown here.

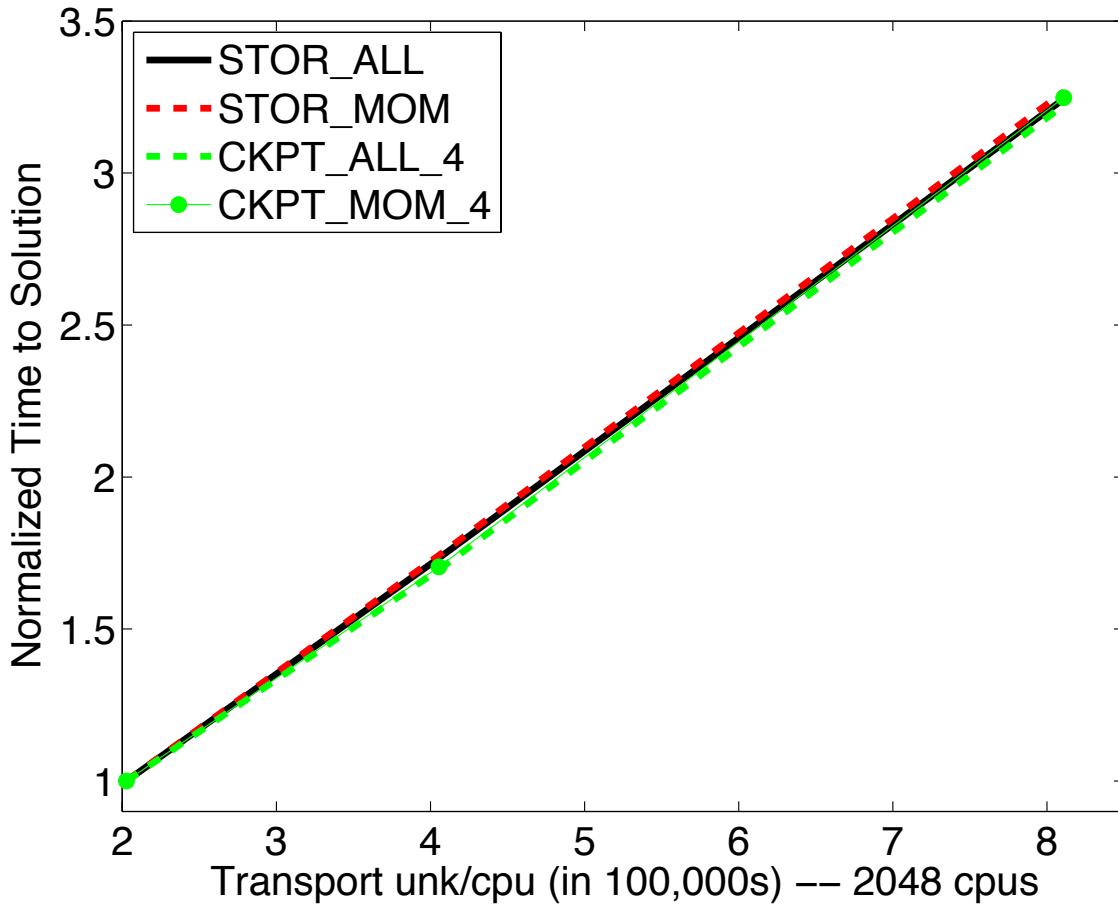


Figure 6.35: Scaling the unknown count for the checkpoint scheme scaling study (2k processors)

The overall takeaway from the scaling study is that the new schemes that store, read, and write only the converged source moments achieve the same (or slightly better) time to solution for this problem but have a significantly lower RAM footprint on the machine. From this, we are confident that the schemes will prove advantageous for larger problems where the STOR_ALL and CKPT_ALL schemes will have too large of a memory footprint. Further ideas to minimize the memory footprint are discussed in the following section. Checkpointing schemes that follow this strategy will facilitate high-fidelity problems on advanced, memory-limited architectures.

7. SUMMARY AND CONCLUSIONS

This section provides a summary of the research findings and contributions of the dissertation. It concludes with closing remarks and suggestions for future research.

7.1 Summary

This dissertation presents the development, application, and implementation of a framework for computing UQSA information using an adjoint approach. We developed the framework in a mathematically rigorous and general manner in order to facilitate a multi-physics modeling environment. We then applied the framework to both the source-driven and k-eigenvalue forms of the depletion equations. The result of this application is a set of adjoint equations which, when solved for a particular problem and quantity of interest, lead to sensitivity and error estimates. Finally, we implemented the framework in the PDT code. We documented a number of verification problems, which are designed in a hierarchical fashion, as well as results from two relevant application problems and one scaling study.

In the following paragraphs, we highlight the major accomplishments of each of these efforts. First, in Sec. 3, we present a variational derivation of the adjoint equations corresponding to a parameter-dependent system of forward differential-algebraic equations. In the spirit of the optimization and control communities, this derivation maintains a very general notation and mathematical rigor. We differ, however, in that we do not make any assumptions about the form of the parameter dependence or the form of the inner product defining the adjoint operators. Ultimately, this allows for adjoint equations in terms of the familiar adjoint transport operator, which has a rich history of application in the nuclear engineering community.

The derivation of the adjoint equations is carried out in a variational manner. From this, we see that the adjoint equations can be interpreted as a set of conditions on a Lagrange multiplier that make first-order variations of the QOI insensitive to first-order perturbations in the system unknowns. The sensitivity equation, which gives the first-order QOI variation in terms of the forward and adjoint solutions, has the form of an adjoint-weighted time-integral of local sensitivities. Likewise, the error equation is an adjoint-weighted time-integral of local truncation errors. Thus, the adjoint variable is often interpreted as a weighting factor.

We apply our general framework to both the source-driven and k-eigenvalue formulations of the depletion equations. The source-driven version is simpler to follow and understand, but the k-eigenvalue equations are most appropriate for application to reactor problems. Common to both versions is the need to have access to the forward solution during the adjoint solve. Because we are interested in time-dependent problems, this poses a formidable computational challenge for large problems because machine memory limitations prohibit the storage of the full time-series of the forward solution. This prompts the introduction of checkpointing schemes, which write snapshots of the forward solution to files during an initial forward solve, then recompute the forward solution in smaller intervals for use during the adjoint solve.

The implementation of solvers for both the forward and adjoint depletion equations is a major contribution of this dissertation. The solvers build on the capabilities of the PDT code, which has demonstrated efficient scaling performance to hundreds of thousands of processors. The hierarchy of solvers is described in Sec. 4 and documented in detail in Appendix B.

The most notable contribution of the implementation is the development and testing of a new family of checkpointing schemes that reduce the memory and I/O cost of managing the angular flux solution. The new schemes store, read from file, and

write to file converged source moments instead of the flux vector. The dimensionality of the source moments is often smaller than that of the flux vector by a factor of $\mathcal{O}(10)$ or more, thus saving on the total amount of data that must be stored or written to file. The trade-off for this reduction is an increase in the FLOP cost: we must perform single transport sweeps to recover the flux vector from the source moments whenever it is needed. Our argument is that FLOPs are becoming cheaper and memory is becoming more expensive, relatively. If the reduction in memory cost is larger than the increased FLOP cost, these schemes will reduce time to solution. We present and analyze the costs of these schemes in Sec. 4.6.

Section 5 documents a number of verification problems used to test our PDT implementation. The verification tests are designed in a hierarchical fashion, meaning that we first test the depletion solver, then the transport-only adjoint solver, and then the depletion perturbation solver. This approach was efficient for finding and eliminating code bugs. Some verification problems used analytic reference solutions while others compared adjoint sensitivities to finite difference estimates. While analytic solutions are preferable, they are often impossible to develop for the fully non-linear depletion equations.

We began the verification process with depletion-only problems. One problem uses an infinite medium solution to test convergence rates of the Runge-Kutta time discretization schemes. In a second problem, we compare the depletion solution of a 33-group, brick reactor problem to results generated by the Matlab ODE suite. The second set of problems test the transport-only adjoint solver, or the steady-state adjoint solver. Again, problem one compares adjoint-based sensitivities to analytic sensitivities for a simple problem, and problem two compares adjoint-based parameter derivatives to those estimated by brute-force parameter perturbations and finite difference. Finally, the last and main verification problem tests the depletion pertur-

bation solver for a k-eigenvalue problem in an infinite medium. We show a number of results, including convergence towards analytic solutions for both sensitivity and error estimates, as well as agreement with expected numerical results via a step-by-step walkthrough of the solution procedure.

Although we would never claim that the code is bug-free, the rigorous verification procedure that we document provided the means to explore the behaviors of all of the new solvers and facilitated the development of a standard set of test-problems that must pass before additions or changes to the code can be admitted.

Finally, in Sec. 6, we describe the application of our UQSA methodology to larger reactor problems that are relevant to ongoing research efforts. First, we attack a depletion benchmark problem modeling a traveling-wave reactor. We use a 33-group model with 18 actinides and 13 lumped fission products. Our solution is similar to the only available, published benchmark solution, and we are confident that the difference between the solution is related to the number of nuclides tracked and the nuclear data source (multigroup cross sections) used. In addition to the specified benchmark solutions, we provide results from simulated UQ studies where we use the adjoint capability to identify driving uncertain inputs, characterize their effect on key quantities of interest, and estimate the error in the QOI due to time discretization.

The second application problem is related to depletion and calibration of a model of the Nuclear Science Center reactor on the campus of Texas A&M. We are interested in developing a model for the reactor over its operational lifetime since refueling in 2006. This requires depletion of the fresh fuel at fairly high fidelity and the calibration of cross-section and temperature dependence models for the Zirconium-Hydride moderator in the fuel. Many of these capabilities are under development or just coming online; therefore, we provide an example 1-year depletion solution that makes physical sense, as well as a framework that simulates the calibration of poison

concentrations to match BOL conditions.

The last major section of the dissertation describes a scaling study for characterizing the performance of our new checkpointing algorithms. We scale in terms of number of processors and in terms of the number of unknowns per processor. Our conclusion is that the new schemes that read, write, and store only the converged transport source moments are computationally advantageous compared to classic checkpointing schemes. We project that these schemes will allow for high-fidelity transport solutions on memory-limited architectures.

7.2 Conclusions

Depletion perturbation with full transport solutions is a tractable problem on advanced computers for realistic, reactor analysis calculations. In this dissertation, we showed that the adjoint equations corresponding to source-driven and k-eigenvalue equations can be derived using a general, mathematical framework, and that the adjoint solutions provide both sensitivity and error estimates for reactor QOIs. We also showed that the cost of producing and managing access to fully angular dependent flux solutions can be managed using a combination of efficient transport solvers and careful schemes for representing the angular flux in memory and in files. We demonstrated our capability on large core counts, showing promising scaling results, as well as for relevant reactor analysis calculations. Thus, we believe that core-level, high-fidelity calculations with adjoint capability are achievable with the PDT code.

Moving forward, we are aware of the challenges that oppose the implementation of the adjoint approach in scientific software. First, we note that the adjoint technique is code-intrusive, meaning that significant source-code access and manipulation is required to both manage the forward solution and propagate (compute) the adjoint solution. Thus, for existing and established system or legacy codes that do not have

this flexibility, implementation of this adjoint technique may be impossible. Another challenge in the case of multiphysics simulations is that the number of functional derivatives (Jacobians) that must be computed and supplied to the adjoint solver can grow rapidly with the number of physical phenomena being modeled. It is possible that advances in automatic differentiation technology[13, 60] will ease the burden of computing these derivatives, a subject that we may consider in the future. On the other hand, the addition of new physics likely introduces a large number of new parameters to which the forward QOIs may be sensitive, further justifying the effort to implement the adjoint technique.

We see a number of avenues for future research. Further characterization of the checkpointing schemes, including performance modeling on particular machines and fine-tuning the file I/O procedures, is an open area of research. In the current implementation, each processor reads and writes its own files to disk. Reduction schemes, such as a reduction to $\mathcal{O}\sqrt{P}$ cores that read/write information to disk may be appropriate depending on the size of P and the host architecture. Also, a performance model to predict the optimal checkpointing frequency would be useful, as the characterizations made in this dissertation were mostly heuristic. Such a performance model will depend on the number of unknowns per processor and the available memory for checkpointing the solution.

We also see possibilities to modify and extend the checkpointing schemes to further reduce their memory footprint. The schemes described in this dissertation require at least two allocations for the full angular flux vector: one for the forward solution and one for the adjoint solution. For some problems on some machines, even this amount of memory may not be available in RAM. One option to reduce this footprint is to read, write, and store only single precision copies of the flux. This will reduce the footprint by a factor of two at the cost of some loss of accuracy.

This source of error, however, will only become significant if the iterative and solver tolerances are on the order of 10^{-7} , which is not always required for reactor analysis calculations.

Another possible extension of the checkpointing schemes is to reduce the memory footprint by performing group-by-group sweeps. In other words, instead of sweeping the domain to recover all groups in the angular flux solution, we could perform the sweep for only a single group or subset of groups, store those entries in the angular flux vector, use them to compute the terms that we need, and then repeat for the next subset of groups. This is a viable strategy because the forward angular flux solution most often appears in energy-integrated or angle-integrated form in the adjoint equations; therefore, each sweep for a group subset would just act to accumulate these integrals. In PDT, this could be implemented by simply reducing the number of group calculations in the inner loop for a given cell and direction.

There is also room for future research in the adjoint-based error estimation. We outlined our procedure for producing time-discretization error estimates, showed super-optimal convergence of this estimate for a simple problem, and showed that our estimate agreed with a Richardson extrapolation error estimate at minimal extra cost. We did not address error in the other phase space variables. For example, error in the choice of quadrature order and number of energy groups is likely more dominant than time-discretization. Ongoing work at Texas A&M involves local error estimators for these variables; these estimates should be integrated with the adjoint framework to produce global error estimates for time-dependent problems.

The implementation described here only dealt with terminal QOIs, or QOIs that were dependent only on the forward solution at $t = t_f$. Our analysis details the steps necessary to handle time-integrated QOIs, and these will likely be needed for the NSC depletion calculations in the near future. Implementation of these

QOIs should be straightforward in the classes and methods we developed in PDT. Thermal feedback modeling will also be important for characterization of the NSC reactor during transient operations. PDT does not currently handle temperature dependence of nuclear cross sections; if this capability is added, careful work will be required to properly implement the adjoint of this feedback for sensitivity studies.

Finally, the behavior of the framework as a whole should continue to be characterized. This dissertation covered only a small subset of the possible applications of the adjoint capability; it is well known, however, that each adjoint problem will behave differently and may have stiffness and stability issues that are not apparent during the forward calculation. The modeler must have an understanding of the dynamics involved with the adjoint equations in order to ensure that the depletion and depletion perturbation functionality is applied correctly in new or different research domains.

REFERENCES

- [1] W. Hawkins, T. Smith, M. Adams, L. Rauchwerger, N. Amato, and M. Adams, “Efficient massively parallel transport sweeps,” *Trans. Amer. Nucl. Soc.*, vol. 107, pp. 477–481, 2012.
- [2] M. Adams, M. Adams, W. Hawkins, T. Smith, L. Rauchwerger, N. Amato, T. Bailey, and R. Falgout, “Provably optimal parallel transport sweeps on regular grids,” *Proc. Intl. Conf. Math. Comp. Meth. Nuc. Sci. Engr.*, 2013.
- [3] The National Research Council: Committee of Mathematical Foundations of Verification Validation and Uncertainty Quantification; Board on Mathematical Sciences and Their Applications; Division on Engineering and Physical Sciences., “Assessing the reliability of complex models: Mathematical and statistical foundations of verification, validation, and uncertainty quantification,” tech. rep., The National Academy of Sciences, 2012.
- [4] The Advanced Scientific Computing Advisory Committee, “The opportunities and challenges of exascale computing,” tech. rep., ASCR, 2010.
- [5] K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, J. Hiller, S. Karp, S. Keckler, D. Klein, R. Lucas, M. Richards, A. Scarpelli, S. Scott, A. Snavely, T. Sterling, R. S. Williams, K. Yelick, K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, J. Hiller, S. Keckler, D. Klein, P. Kogge, R. S. Williams, and K. Yelick, “ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems Peter Kogge, Editor and Study Lead.,” 2008. tech. rep., United States Department of Energy.

- [6] M. D. DeHart, *TRITON: A two-dimensional transport and depletion module for characterization of spent nuclear fuel*. Oak Ridge National Laboratory, ORNL/TM-2005/39, 2005. Revision 5.1 Vol. I, Book 3, Sect. T1.
- [7] J. S. Hendricks, G. W. McKinney, and M. L. Fensin, *MCNPX, version 26e*, 2007. Los Alamos Report, LA-UR-07-6632, November 2007.
- [8] D. I. Poston and H. R. Trellue, *Development of a Fully Automated Monte Carlo Burnup Code MONTEBURNS*, 1999. Los Alamos Report, LA-UR-99-42, January 1999.
- [9] S. Dulla, E. H. Mund, and P. Ravetto, “The quasi-static method revisited,” *Prog. Nucl. Energy*, vol. 50, no. 8, pp. 908–920, 2008. In Honour of Prof. Bruno Montagnini.
- [10] G. Bell and S. Glasstone, *Nuclear Reactor Theory*. Van Nostrand Reinhold, New York, 1970.
- [11] A. Henry, *Nuclear Reactor Analysis*. MIT Press, Cambridge, 1975.
- [12] W. S. Yang and T. J. Downar, “A linear time-dependent flux approximation for nuclide depletion based on a perturbation method,” *Ann. Nuc. Energy*, vol. 17, no. 1, pp. 37–42, 1990.
- [13] A. Griewank and A. Walther, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2008.
- [14] P. Kunkel and V. Mehrmann, *Differential-Algebraic Equations: Analysis and Numerical Solution*. European Mathematical Society, Zurich, 2006.
- [15] D. Kincaid and W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, 3rd Ed. Brooks/Cole, Pacific Grove CA, 2002.

- [16] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer-Verlag, Berlin, 1996.
- [17] E. Hairer and M. Roche, *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*. Springer-Verlag, Berlin, 1989.
- [18] U. Ascher and L. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, ch. 10. Philadelphia: Society for Industrial and Applied Mathematics, 1998.
- [19] Y. Cao, S. Li, L. Petzold, and R. Serban, “Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution,” *SIAM J Comput. Appl. Math.*, vol. 24, no. 3, pp. 1076–1089, 2003.
- [20] E. Lewis and W. Miller, *Computational Methods of Neutron Transport*. The American Nuclear Society, La Grange, IL, 1993.
- [21] R. Hartmann, “Adjoint consistency analysis of discontinuous Galerkin discretizations,” *SIAM J Numer. Anal.*, vol. 45, no. 6, pp. 2671–2696, 2007.
- [22] W. Hager, “Runge-kutta methods in optimal control and the transformed adjoint system,” *Numer. Math.*, vol. 87, no. 2, pp. 247–282, 2000.
- [23] A. Sandu, “On the properties of runge-kutta discrete adjoints,” *Lec. Notes in Comp. Sci.*, vol. 3994, pp. 550–557, 2006.
- [24] W. Hager, “Rates of convergence for discrete approximations to unconstrained control problems,” *SIAM J. Numer. Anal.*, vol. 13, no. 4, pp. 449–472, 1976.
- [25] E. Wigner, “Effect of small perturbations on pile period.,” tech. rep., Manhattan Project, 1945.
- [26] G. Pomraning, “A variational principle for linear systems,” *J. Soc. Indust. Appl. Math.*, vol. 13, no. 2, pp. 511–519, 1965.

- [27] J. Lewins, *Importance: the adjoint function. The physical basis of variational and perturbation theory in transport and diffusion problems*. Pergamon, Oxford, 1965.
- [28] A. Gandini, “A generalized perturbation method for bi-linear functionals of the real and adjoint neutron fluxes,” *J. of Nuc. Energy*, vol. 21, pp. 755–765, 1967.
- [29] W. M. Stacey, *Variational Methods in Nuclear Reactor Physics*. Academic Press, New York, 1974.
- [30] D. Cacuci, “Sensitivity theory for nonlinear systems. I. nonlinear functional analysis approach,” *J. Math. Phys.*, vol. 22, no. 12, pp. 2794–2802, 1981.
- [31] D. Cacuci, “Sensitivity theory for nonlinear systems. II. extensions to additional classes of responses,” *J. Math. Phys.*, vol. 22, no. 12, pp. 2803–2812, 1981.
- [32] M. L. Williams, *Perturbation Theory for Reactor Analysis, CRC Handbook of Nuclear Reactor Calculations*. CRC Press, 1986.
- [33] D. Cacuci, “Global optimization and sensitivity analysis,” *Nuc. Sci. Engr.*, vol. 104, no. 1, pp. 78–88, 1990.
- [34] M. L. Williams and B. T. Rearden, “SCALE-6 sensitivity/uncertainty methods and covariance data,” *Nuc. Data Sheets*, vol. 109, pp. 2796–2800, 2008.
- [35] M. A. Jessee, M. L. Williams, and M. D. Dehart, “Development of generalized perturbation theory capability within the SCALE code package,” *Proc. Intnl. Conf. Math., Comp. Meth., Rctr. Phys.*, 2009.
- [36] W. Wieselquist, A. Vasiliev, and H. Ferroukhi, “Towards an uncertainty quantification methodology with CASMO-5,” *Proc. Intnl. Conf. Math., Comp. Meth., and Rctr. Phys.*, 2011. Rio de Janeiro, Brazil.

- [37] C. Wang and H. S. Abdel-Khalik, “Exact-to-precision generalized perturbation theory for source driven systems,” *Nuc. Eng. and Des.*, vol. 241, pp. 5104–5112, 2011.
- [38] H. S. Abdel-Khalik, “Adjoint-based sensitivity analysis for multi-component models,” *Nuc. Eng. and Des.*, vol. 245, pp. 49–54, 2011.
- [39] M. L. Williams, “Development of depletion perturbation-theory for coupled neutron-nuclide fields,” *Nucl. Sci. Engr.*, vol. 70, no. 1, pp. 20–36, 1979.
- [40] T. Takeda and T. Umano, “Burnup sensitivity analysis in a fast breeder reactor – part 1: Sensitivity calculation method with generalized perturbation theory,” *Nucl. Sci. Engr.*, vol. 91, pp. 1–10, 1985.
- [41] N. Zavaljevski, “A model for fuel shuffling and burnable absorbers optimization in low leakage PWRs,” *Ann. Nucl. Energy*, vol. 17, no. 4, pp. 217–220, 1990.
- [42] F. C. da Silva and Z. D. Thome, “Depletion calculations with static generalized perturbation theory,” *Annals of Nuclear Energy*, vol. 15, no. 8, pp. 431–434, 1988.
- [43] T. J. Downar, “Depletion perturbation theory for burnup dependent microscopic cross sections,” *Annals of Nuclear Energy*, vol. 19, no. 1, pp. 27–37, 1992.
- [44] 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*, vol. 28, no. 14, pp. 1377–1411, 2001.
- [45] G. Chiba, K. Okumura, A. Oizumi, and M. Saito, “Sensitivity analysis of fission product concentrations for light water reactor burned fuel,” *J Nuc. Sci. Tech.*, vol. 47, no. 7, pp. 652–660, 2010.

- [46] W. S. Yang and T. J. Downar, “Generalized perturbation-theory for constant power core depletion,” *Nuclear Science and Engineering*, vol. 99, no. 4, pp. 353–366, 1988.
- [47] Y. Cao and L. Petzold, “A posteriori error estimation and global error control for ordinary differential equations by the adjoint method,” *SIAM J. Sci. Comput.*, vol. 26, no. 2, pp. 359–374, 2004.
- [48] H. F. Stripling, M. Anitescu, and M. L. Adams, “A generalized adjoint framework for sensitivity and global error estimation in time-dependent nuclear reactor simulations,” *Ann. Nucl. Energy*, vol. 52, pp. 47–58, 2012.
- [49] Y. Cao, S. Li, and L. Petzold, “Adjoint sensitivity analysis for differential algebraic equations: Algorithms and software,” *SIAM J Comput. Appl. Math.*, vol. 149, no. 1, pp. 171–191, 2002.
- [50] E. I. Fredholm, “Sur une classe d’equations fonctionnelles,” *Acta. Math.*, pp. 365–390, 1903.
- [51] D. J. Estep, “A short course on duality, adjoint operators, Green’s functions, and *a posteriori* error analysis,” 2004. tech. rep.
- [52] J. Chang, *Efficient Algorithms for Discrete-Ordinates Transport Iterations in Massively Parallel Computers*. PhD thesis, Texas A&M University, College Station, TX, 2004.
- [53] M. Adams and E. Larsen, “Fast iterative methods for discrete-ordinates particle transport calculations,” *Prog. Nucl. Energy*, vol. 40, no. 3, 2002.
- [54] E. Oblow, “Reactor cross-section sensitivity studies using transport theory,” Tech. Rep. TM-4437, Oak Ridge National Laboratory, 1974.

- [55] L. Shampine and M. Reichelt, “The Matbal ODE suite,” *SIAM J Sci. Comp.*, vol. 18, no. 1, pp. 1–22, 1997.
- [56] R. Petroski, *General Analysis of Breed-and-Burn Reactors and Limited-Separations Fuel Cycles*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2011.
- [57] Z. Xu, R. Petroski, N. Touran, and C. Whitmer, “A one-dimensional benchmark problem of breed & burn reactor,” *Trans. of the Amer. Nuc. Soc.*, vol. 104, pp. 786–787, 2011.
- [58] T. Santner, B. Williams, and W. Notz, *The Design and Analysis of Computer Experiments*. Springer-Verlag, New York, 2003.
- [59] G. Marleau, A. Hebert, and R. Roy, “Dragon: A collision probability transport code for cell and multicell calculations,” tech. rep., Ecole Polytechnique de Montreal, 1990.
- [60] A. Griewank and G. F. Corliss, *Automatic differentiation of algorithms: theory, implementation, and application*. Society for industrial and Applied Mathematics Philadelphia, PA, 1991.
- [61] A. Halanay, *Differential Equations: Stability, Oscillations, Time Lags*. Academic Press, Inc, London, 1966.

APPENDIX A

A VARIATIONAL DERIVATION OF THE ADJOINT DEPLETION EQUATIONS CORRESPONDING TO THE SOURCE DRIVEN AND K-EIGENVALUE BURNUP EQUATIONS

This appendix presents a formal derivation of the source-driven and k-eigenvalue depletion perturbation equations using a variational approach. In Sec. 3.1.1, we performed a similar derivation using general operators and a general inner product. The resulting framework was then applied to the two forward cases to form the corresponding adjoint equations. Here, we begin with the specific form of the forward equations and derive, from scratch, the appropriate adjoint equations. These equations are identical to those developed in Secs. 3.2 and 3.3.

A.1 The Source-Driven Case

The source-driven forward problem is

$$\frac{dN}{dt} - B(\psi, p)N = 0, \quad (\text{A.1})$$

$$H(N, p)\psi - G(N, p)\psi - S_0 = 0, \quad (\text{A.2})$$

$$N(t = 0) = N_0(p), \quad t \in [t_0, t_f], \quad (\text{A.3})$$

where B , H , and G are the Bateman, transport, and fission operators defined in Sec. 2.1. We explicitly emphasize the dependence of these operators on a set of input or numerical parameters, p , to which our QOI may be sensitive. For example, p may consist of all microscopic cross sections required to specify the problem.

In Sec. 2.4, we presented the form of adjoint operators that satisfy the following

equations

$$\langle \psi^\dagger H \psi \rangle_{E,\mathcal{D},\Omega} = \langle \psi H^\dagger \psi^\dagger \rangle_{E,\mathcal{D},\Omega}, \quad (\text{A.4})$$

$$\langle \psi^\dagger G \psi \rangle_{E,\mathcal{D},\Omega} = \langle \psi G^\dagger \psi^\dagger \rangle_{E,\mathcal{D},\Omega}, \quad (\text{A.5})$$

$$\langle N^\dagger b N \rangle_{E,\mathcal{D},\Omega} = \langle N b^\dagger N^\dagger \rangle_{E,\mathcal{D},\Omega}. \quad (\text{A.6})$$

Recall that the inner product $\langle \rangle_{E,\mathcal{D},\Omega}$ is an integration over phase space, or angle, energy, and volume.

To begin, we assume that our QOI is a time-integrated functional R of the depletion solution, namely

$$Q = \int_{t_0}^{t_f} \langle R(N, \psi, p, t) \rangle_{E,\mathcal{D},\Omega} dt. \quad (\text{A.7})$$

We derive the adjoint equations as follows. We first take the inner product of each of our governing equations multiplied by its corresponding adjoint vector and write

$$\begin{aligned} N^\dagger \left(\frac{dN}{dt} - \langle bN \rangle_{E,\mathcal{D},\Omega} \right) &= 0 \\ \langle \psi^\dagger, H\psi - G\psi - S_0 \rangle_{E,\mathcal{D},\Omega} &= 0 \end{aligned}$$

We then subtract (*adjoin*) each of these terms from Eq. (A.7) to form a Lagrangian, \mathcal{L} :

$$\begin{aligned} \mathcal{L} &= \int_{t_0}^{t_f} \langle R(N, \psi, p, t) \rangle_{E,\mathcal{D},\Omega} dt - \int_{t_0}^{t_f} \langle \psi^\dagger, H\psi - G\psi - S_0 \rangle_{E,\mathcal{D},\Omega} dt \\ &\quad - \int_{t_0}^{t_f} N^\dagger \left(\frac{dN}{dt} - \langle bN \rangle_{E,\mathcal{D},\Omega} \right) dt. \end{aligned} \quad (\text{A.8})$$

Note that if N and ψ satisfy the forward equations, then $\mathcal{L} = Q$ and $\frac{d\mathcal{L}}{dp} = \frac{dQ}{dp}$. We apply the chain rule to take the full derivative $\frac{d\mathcal{L}}{dp}$ of the Lagrangian with respect to the vector of parameters. In the following expression, we have applied Eqs. (A.4), (A.5), and (A.6) strategically.

$$\begin{aligned}
\frac{d\mathcal{L}}{dp} = & \int_{t_0}^{t_f} \left(\frac{\partial}{\partial \psi} \langle R \rangle_{E,\mathcal{D},\Omega} \frac{d\psi}{dp} \right) dt + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial N} \langle R \rangle_{E,\mathcal{D},\Omega} \frac{dN}{dp} \right) dt + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial p} \langle R \rangle_{E,\mathcal{D},\Omega} \right) dt \\
& - \int_{t_0}^{t_f} \left(\frac{\partial}{\partial \psi} \langle \psi, H^\dagger \psi^\dagger \rangle_{E,\mathcal{D},\Omega} \frac{d\psi}{dp} \right) dt - \int_{t_0}^{t_f} \left(\frac{\partial}{\partial N} \langle \psi^\dagger, H\psi \rangle_{E,\mathcal{D},\Omega} \frac{dN}{dp} \right) dt \\
& \quad - \int_{t_0}^{t_f} \left(\frac{\partial}{\partial p} \langle \psi^\dagger, H\psi \rangle_{E,\mathcal{D},\Omega} \right) dt \\
& + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial \psi} \langle \psi, G^\dagger \psi^\dagger \rangle_{E,\mathcal{D},\Omega} \frac{d\psi}{dp} \right) dt + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial N} \langle \psi^\dagger, G\psi \rangle_{E,\mathcal{D},\Omega} \frac{dN}{dp} \right) dt \\
& \quad + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial p} \langle \psi^\dagger, G\psi \rangle_{E,\mathcal{D},\Omega} \right) dt \\
& + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial p} \langle \psi^\dagger, S_0 \rangle_{E,\mathcal{D},\Omega} \right) dt \\
& + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial \psi} \left[N \frac{dN^\dagger}{dt} \right] \frac{d\psi}{dp} \right) dt + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial N} \left[N \frac{dN^\dagger}{dt} \right] \frac{dN}{dp} \right) dt + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial p} \left[N \frac{dN^\dagger}{dt} \right] \right) dt \\
& + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial \psi} \langle N^\dagger bN \rangle_{E,\mathcal{D},\Omega} \frac{d\psi}{dp} \right) dt + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial N} \langle Nb^\dagger N^\dagger \rangle_{E,\mathcal{D},\Omega} \frac{dN}{dp} \right) dt \\
& \quad + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial p} \langle N^\dagger bN \rangle_{E,\mathcal{D},\Omega} \right) dt \\
& - \left[\frac{\partial}{\partial N} \left[N^\dagger N \right] \frac{dN}{dp} \right]_{t_0}^{t_f}. \tag{A.9}
\end{aligned}$$

Many of the terms in Eq. (A.9) are known or can be derived, including many of the partial derivatives. However, the full solution Jacobians $\frac{d\psi}{dp}$ and $\frac{dN}{dp}$ that appear in nearly every term are incomputable (in fact, if we knew these, we would have straightforward expressions for $\frac{dQ}{dp}$ and we'd be done). The adjoint equations pop out via the strategy to eliminate these terms by setting their coefficients to zero. We can re-write Eq. (A.9) by combining those terms that multiply the full solution

Jacobians and applying partial derivatives wherever possible:

$$\begin{aligned}
\frac{d\mathcal{L}}{dp} = & \int_{t_0}^{t_f} \left(\left[\left\langle \frac{\partial R}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle H^\dagger \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} + \left\langle G^\dagger \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} + \left\langle N^\dagger \frac{\partial bN}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} \right] \frac{d\psi}{dp} \right) dt \\
& + \int_{t_0}^{t_f} \left(\left[\left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial H\psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \psi^\dagger, \frac{\partial G\psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} \right. \right. \\
& \quad \left. \left. + \frac{dN^\dagger}{dt} + \left\langle b^\dagger N^\dagger \right\rangle_{E,\mathcal{D},\Omega} \right] \frac{dN}{dp} \right) dt \\
& + \int_{t_0}^{t_f} \left(\left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial H\psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \psi^\dagger, \frac{\partial G\psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \right. \\
& \quad \left. + \left\langle \psi^\dagger, \frac{\partial S_0}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle N^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \right) dt \\
& - \left[N^\dagger \frac{dN}{dp} \right]_{t=t_f} + \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0}
\end{aligned} \tag{A.10}$$

Now note that if these equations are satisfied,

$$\begin{aligned}
H^\dagger \psi^\dagger - G^\dagger \psi^\dagger &= \frac{\partial R}{\partial \psi} + N^\dagger \frac{\partial bN}{\partial \psi} \\
\frac{dN^\dagger}{dt} &= \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - G\psi] \right\rangle_{E,\mathcal{D},\Omega} - B^\dagger N^\dagger - \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega}
\end{aligned}$$

then Eq. (A.10) reduces to

$$\begin{aligned}
\frac{d\mathcal{L}}{dp} = & \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0} - \left[N^\dagger \frac{dN}{dp} \right]_{t=t_f} \\
& + \int_{t_0}^{t_f} \left(\left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - G\psi - S_0] \right\rangle_{E,\mathcal{D},\Omega} \right. \\
& \quad \left. + \left\langle N^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \right) dt
\end{aligned} \tag{A.11}$$

We have now eliminated the full solution Jacobians from the integral terms; we have yet to eliminate $\frac{dN}{dp}$ at $t = t_0$ and $t = t_f$. For the $t = t_f$ case, we simply impose a

terminal condition on the adjoint system, namely

$$N^\dagger(t_f) = 0.$$

The implication of this terminal condition is that the adjoint equations must be solved backwards in time. The term $\frac{dN}{dp}$ at $t = t_0$ requires that we know the full gradient of the initial condition with respect to the list of parameters. As mentioned before, we assume that this information is either known (via some explicit function) or given (maybe by another code). A third option is that the parameters are the initial conditions themselves, giving $\frac{dN}{dP} = 1$ and providing an avenue for computing the sensitivity of the QOI with respect to the initial densities. No matter the case, we leave this term in the sensitivity equation and assume it is computable.

Under these conditions, Eq. (A.11) becomes

$$\begin{aligned} \frac{d\mathcal{L}}{dp} &= \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0} + \int_{t_0}^{t_f} \left(\left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - G\psi - S_0] \right\rangle_{E,\mathcal{D},\Omega} \right. \\ &\quad \left. + \left\langle N^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \right) dt \\ &= \frac{dQ}{dp} \end{aligned}$$

In summary, for the case of a time-integrated QOI, we found adjoint equations

$$H^\dagger \psi^\dagger - G^\dagger \psi^\dagger = \frac{\partial \mathcal{L}}{\partial \psi} + N^\dagger \frac{\partial bN}{\partial \psi} \tag{A.12}$$

$$\frac{dN^\dagger}{dt} = \left\langle \psi^\dagger \frac{\partial H\psi}{\partial N} - \psi^\dagger \frac{\partial G\psi}{\partial N} - b^\dagger N^\dagger - \frac{\partial \mathcal{L}}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} \tag{A.13}$$

with terminal condition for the adjoint densities

$$N^\dagger(t_f) = 0. \quad (\text{A.14})$$

If N^\dagger and ψ^\dagger satisfy this system, then the gradient of the time integrated QOI w.r.t. the parameter vector is

$$\begin{aligned} \frac{dQ}{dp} = & \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0} + \int_{t_0}^{t_f} \left(\left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - G\psi - S_0] \right\rangle_{E,\mathcal{D},\Omega} \right. \\ & \left. + \left\langle N^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \right) dt \end{aligned} \quad (\text{A.15})$$

Note: Eqs. (A.12)–(A.15) are identical to Eqs. (3.43)–(3.46).

We now consider the case of a terminal QOI, or a QOI that depends only on the solution at $t = t_f$:

$$Q_f = \left\langle R(N(t_f), \psi(t_f), p, t_f) \right\rangle_{E,\mathcal{D},\Omega}. \quad (\text{A.16})$$

The derivation begins by noting that

$$\frac{dQ_f}{dp} = \frac{d}{dt_f} \frac{dQ}{dp}.$$

Also, recall the following rule of calculus:

$$\frac{d}{dt_f} \int_{t_0}^{t_f} f(t, t_f) dt = f(t_f, t_f) + \int_{t_0}^{t_f} \frac{df}{dt_f} dt,$$

which we apply to Eq. (A.10):

$$\begin{aligned}
\frac{d\mathcal{L}_f}{dp} = & \left[\left[\left\langle \frac{\partial R}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle H^\dagger \psi^\dagger - G^\dagger \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} + \left\langle N^\dagger \frac{\partial bN}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} \right] \frac{d\psi}{dp} \right]_{t=t_f} \\
& + \left[\left[\left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right. \right. \\
& \quad \left. \left. + \frac{dN^\dagger}{dt} + \left\langle b^\dagger N^\dagger \right\rangle_{E,\mathcal{D},\Omega} \right] \frac{dN}{dp} \right]_{t=t_f} \\
& + \left[\left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - G\psi - S_0] \right\rangle_{E,\mathcal{D},\Omega} + \left\langle N^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \right]_{t=t_f} \\
& + \left[N_f^\dagger \frac{dN}{dp} \right]_{t=t_0} \\
& + \int_{t_0}^{t_f} \left(\left[- \left\langle H^\dagger \psi_f^\dagger - G^\dagger \psi_f^\dagger \right\rangle_{E,\mathcal{D},\Omega} + \left\langle N_f^\dagger \frac{\partial bN}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} \right] \frac{d\psi}{dp} \right) dt \\
& + \int_{t_0}^{t_f} \left[\left[- \left\langle \psi_f^\dagger, \frac{\partial}{\partial N} [H\psi - G\psi] \right\rangle_{E,\mathcal{D},\Omega} + \frac{dN_f^\dagger}{dt} \right. \right. \\
& \quad \left. \left. + \left\langle b^\dagger N_f^\dagger \right\rangle_{E,\mathcal{D},\Omega} \right] \frac{\partial N}{\partial t} \right] dt \\
& + \int_{t_0}^{t_f} \left(- \left\langle \psi_f^\dagger, \frac{\partial}{\partial p} [H\psi - G\psi - S_0] \right\rangle_{E,\mathcal{D},\Omega} + \left\langle N_f^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} \right) dt
\end{aligned} \tag{A.17}$$

Here N_f^\dagger and ψ_f^\dagger are $\frac{\partial N^\dagger}{\partial t_f}$ and $\frac{\partial \psi^\dagger}{\partial t_f}$ respectively, as they are functions of both t and t_f .

The adjoint equations are extracted from Eq. (A.17) in order to eliminate $\frac{dN}{dp}$ and $\frac{d\psi}{dp}$ from the integral terms. They are

$$\begin{aligned}
\frac{dN_f^\dagger}{dt} &= -B^\dagger N_f^\dagger + \left\langle \psi_f^\dagger, \frac{\partial}{\partial N} [H\psi - G\psi] \right\rangle_{E,\mathcal{D},\Omega} \\
H^\dagger \psi_f^\dagger - G^\dagger \psi_f^\dagger &= N_f^\dagger \frac{\partial bN}{\partial \psi}.
\end{aligned}$$

We use the terminal condition for N_f^\dagger to eliminate the terms $\frac{d\psi}{dp}$ and $\frac{dN}{dp}$ that appear

at $t = t_f$. First, recall that our terminal condition for N^\dagger is

$$N^\dagger(t_f) = 0.$$

Taking the full derivative with respect to time, we find

$$N_f^\dagger(t_f) + \frac{dN^\dagger}{dt}\Big|_{t=t_f} = 0.$$

To eliminate $\frac{dN}{dp}\Big|_{t=t_f}$ from Eq. (A.17), we need

$$\begin{aligned} \frac{dN^\dagger}{dt}\Big|_{t=t_f} &= \left[\left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - G\psi] \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - B^\dagger N^\dagger \right]_{t=t_f} \\ &= \left[\left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - G\psi] \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} \right]_{t=t_f}. \end{aligned}$$

Therefore, the terminal condition for N_f^\dagger is

$$N_f^\dagger(t_f) = \left[\left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right]_{t=t_f}.$$

Now we just need to define $\psi^\dagger(t_f)$, which appears in the above equation. It is set to eliminate $\frac{d\psi}{dp}\Big|_{t=t_f}$ from Eq. (A.17). That is, it must satisfy

$$H^\dagger \psi^\dagger - G^\dagger \psi^\dagger = \frac{\partial R}{\partial \psi},$$

where we have again accounted for $N^\dagger(t_f) = 0$. To review, we have found that in the case of a terminal QOI, the adjoint variable is really a partial derivative with respect

to the terminal time of the previous adjoint variable. The adjoint equations are

$$\frac{dN_f^\dagger}{dt} = -B^\dagger N_f^\dagger + \left\langle \psi_f^\dagger, \frac{\partial}{\partial N} [H\psi - G\psi] \right\rangle_{E,\mathcal{D},\Omega} \quad (\text{A.18})$$

$$H^\dagger \psi_f^\dagger - G^\dagger \psi_f^\dagger = N_f^\dagger \frac{\partial bN}{\partial \psi}. \quad (\text{A.19})$$

with terminal condition

$$N_f^\dagger(t_f) = \left[\left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right]_{t=t_f} \quad (\text{A.20})$$

where ψ^\dagger satisfies

$$H^\dagger \psi^\dagger - G^\dagger \psi^\dagger = \frac{\partial R}{\partial \psi}. \quad (\text{A.21})$$

If the adjoint system is satisfied by N_f^\dagger and ψ_f^\dagger , the sensitivity of the terminal QOI (Eq. (A.17)) reduces to

$$\begin{aligned} \frac{dQ_f}{dp} &= \left[\left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - G\psi - S_0] \right\rangle_{E,\mathcal{D},\Omega} \right]_{t=t_f} + \left[N_f^\dagger \frac{dN}{dp} \right]_{t=t_0} \\ &\quad + \int_{t_0}^{t_f} \left(\left\langle N_f^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi_f^\dagger, \frac{\partial}{\partial p} [H\psi - G\psi - S_0] \right\rangle_{E,\mathcal{D},\Omega} \right) dt. \end{aligned} \quad (\text{A.22})$$

Note that Eqs. (A.18)–(A.22) are identical to (3.48)–(3.52), which were developed as a specialization of our abstract adjoint framework.

A.2 The k-Eigenvalue Case

We now derive the analogous adjoint equations for the k-eigenvalue case. The k-eigenvalue problem formulation is:

$$\frac{dN}{dt} - B(\psi, A, p)N = 0, \quad (\text{A.23})$$

$$H(N, p)\psi - \lambda G(N, p)\psi = 0, \quad (\text{A.24})$$

$$AP(N, p)\psi - \mathcal{P}(t) = 0 \quad (\text{A.25})$$

$$N(t = 0) = N_0(p), \quad t \in [t_o, t_f]. \quad (\text{A.26})$$

Here, ψ satisfies the fundamental eigenmode of Eq. (A.24) with eigenvalue $\lambda = \frac{1}{k_{\text{eff}}}$, and the parameter $\mathcal{P}(t)$ is the prescribed power density ($\left[\frac{\text{MeV}}{\text{s}\cdot\text{cm}^3}\right]$) of the reactor. The operator P is used to compute the average power density generated by fission,

$$\begin{aligned} \text{Power Density} &= AP\psi \equiv A \int dV \int dE \int d\Omega \frac{\sum_j N_j(r) \sigma_{f,j}(E) E_{f,j} \psi(r, E, \Omega)}{V_R} \\ &= \frac{A}{V_R} \left\langle \sum_j N_j \sigma_{f,j} E_{f,j} \psi \right\rangle_{E, \mathcal{D}, \Omega} \\ &\equiv \frac{A}{V_R} \left\langle \Sigma_E \psi \right\rangle_{E, \mathcal{D}, \Omega} \\ &\equiv \frac{A}{V_R} \hat{\mathcal{P}} \end{aligned}$$

where $E_{f,j}$ is the energy released in a fission event by nuclide j , V_R is the reactor volume, and the symbols Σ_E and $\hat{\mathcal{P}}$ will be used as a short-hand for an energy-production macroscopic cross section and un-normalized power, respectively. Finally, the scalar normalization factor A is the value by which the eigenvector must be scaled to satisfy the power constraint.

Recall the bilinear relationship between ψ and A as they appear in the Bateman

operator. This led to Eqs. (3.58) and (3.59), which are reproduced here:

$$\frac{\partial B}{\partial \psi} = \frac{\partial B}{\partial \Psi} \frac{\partial \Psi}{\partial \psi} = \frac{\partial B}{\partial \Psi} A \quad (\text{A.27})$$

$$\frac{\partial B}{\partial A} = \frac{\partial B}{\partial \Psi} \frac{\partial \Psi}{\partial A} = \frac{\partial B}{\partial \Psi} \psi. \quad (\text{A.28})$$

We introduce a new multiplier, A^\dagger , for the power equation, and proceed much the same as before. Beginning with a time-integrated QOI, we form the Lagrangian

$$\begin{aligned} \mathcal{L} = & \int_{t_0}^{t_f} \left\langle R(N, \psi, A, p, t) \right\rangle_{E, \mathcal{D}, \Omega} dt - \int_{t_0}^{t_f} \left\langle \psi^\dagger, H\psi - \lambda G\psi \right\rangle_{E, \mathcal{D}, \Omega} dt \\ & - \int_{t_0}^{t_f} N^\dagger \left(\frac{dN}{dt} - \left\langle bN \right\rangle_{E, \mathcal{D}, \Omega} \right) dt - \int_{t_0}^{t_f} A^\dagger \left(\frac{A}{V_R} \left\langle \Sigma_E \psi \right\rangle_{E, \mathcal{D}, \Omega} - \mathcal{P}(t) \right) dt. \end{aligned} \quad (\text{A.29})$$

We next take the first variation of the Lagrangian (that is, its first derivative w.r.t. our parameter vector). Using the chain rule, this is written as

$$\frac{d\mathcal{L}}{dp} = \frac{\partial \mathcal{L}}{\partial p} + \frac{\partial \mathcal{L}}{\partial \psi} \frac{d\psi}{dp} + \frac{\partial \mathcal{L}}{\partial N} \frac{dN}{dp} + \frac{\partial \mathcal{L}}{\partial \lambda} \frac{d\lambda}{dp} + \frac{\partial \mathcal{L}}{\partial A} \frac{dA}{dp}. \quad (\text{A.30})$$

The partial derivatives in this equation are straightforward to evaluate: they are Jacobians of the discrete transport, fission, Bateman, and power operators. On the contrary, the total derivatives or full variations of the unknowns with respect to the parameters are not readily available. Even for simple test problems, these expressions are complicated by the coupling between the neutron and nuclide equations; for practical problems of interest, a closed form expression for these derivatives is hopeless.

Instead, we proceed by expanding the expression and using stationarity conditions to eliminate the terms we cannot compute. The expanded Lagrangian derivative is

$$\begin{aligned}
\frac{d\mathcal{L}}{dp} = & \int_{t_0}^{t_f} \left(\frac{\partial}{\partial \psi} \langle R \rangle_{E,\mathcal{D},\Omega} \frac{d\psi}{dp} \right) dt + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial N} \langle R \rangle_{E,\mathcal{D},\Omega} \frac{dN}{dp} \right) dt \\
& + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial A} \langle R \rangle_{E,\mathcal{D},\Omega} \frac{dA}{dp} \right) dt + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial p} \langle R \rangle_{E,\mathcal{D},\Omega} \right) dt \\
& - \int_{t_0}^{t_f} \left(\frac{\partial}{\partial \psi} \langle \psi, H^\dagger \psi^\dagger \rangle_{E,\mathcal{D},\Omega} \frac{d\psi}{dp} \right) dt - \int_{t_0}^{t_f} \left(\frac{\partial}{\partial N} \langle \psi^\dagger, H\psi \rangle_{E,\mathcal{D},\Omega} \frac{dN}{dp} \right) dt \\
& - \int_{t_0}^{t_f} \left(\frac{\partial}{\partial A} \langle \psi^\dagger, H\psi \rangle_{E,\mathcal{D},\Omega} \frac{dA}{dp} \right) dt - \int_{t_0}^{t_f} \left(\frac{\partial}{\partial p} \langle \psi^\dagger, H\psi \rangle_{E,\mathcal{D},\Omega} \right) dt \\
& + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial \lambda} \lambda \langle \psi^\dagger G\psi \rangle_{E,\mathcal{D},\Omega} \frac{d\lambda}{dp} \right) dt \\
& + \int_{t_0}^{t_f} \left(\lambda \frac{\partial}{\partial \psi} \langle \psi, G^\dagger \psi^\dagger \rangle_{E,\mathcal{D},\Omega} \frac{d\psi}{dp} \right) dt + \int_{t_0}^{t_f} \left(\lambda \frac{\partial}{\partial N} \langle \psi^\dagger, G\psi \rangle_{E,\mathcal{D},\Omega} \frac{dN}{dp} \right) dt \\
& + \int_{t_0}^{t_f} \left(\lambda \frac{\partial}{\partial A} \langle \psi^\dagger, G\psi \rangle_{E,\mathcal{D},\Omega} \frac{dA}{dp} \right) dt + \int_{t_0}^{t_f} \left(\lambda \frac{\partial}{\partial p} \langle \psi^\dagger, G\psi \rangle_{E,\mathcal{D},\Omega} \right) dt \\
& + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial \psi} \left[N \frac{dN^\dagger}{dt} \right] \frac{d\psi}{dp} \right) dt + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial N} \left[N \frac{dN^\dagger}{dt} \right] \frac{dN}{dp} \right) dt \\
& + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial A} \left[N \frac{dN^\dagger}{dt} \right] \frac{dA}{dp} \right) dt + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial p} \left[N \frac{dN^\dagger}{dt} \right] \right) dt \\
& + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial \psi} \langle N^\dagger bN \rangle_{E,\mathcal{D},\Omega} \frac{d\psi}{dp} \right) dt + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial N} \langle Nb^\dagger N^\dagger \rangle_{E,\mathcal{D},\Omega} \frac{dN}{dp} \right) dt \\
& + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial A} \langle N^\dagger bN \rangle_{E,\mathcal{D},\Omega} \frac{dA}{dp} \right) dt + \int_{t_0}^{t_f} \left(\frac{\partial}{\partial p} \langle N^\dagger bN \rangle_{E,\mathcal{D},\Omega} \right) dt \\
& - \int_{t_0}^{t_f} \left(\frac{\partial}{\partial \psi} \left[A^\dagger \frac{A}{V_R} \hat{\mathcal{P}} - A^\dagger \mathcal{P}(t) \right] \frac{d\psi}{dp} \right) dt \\
& - \int_{t_0}^{t_f} \left(\frac{\partial}{\partial N} \left[A^\dagger \frac{A}{V_R} \hat{\mathcal{P}} - A^\dagger \mathcal{P}(t) \right] \frac{dN}{dp} \right) dt \\
& - \int_{t_0}^{t_f} \left(\frac{\partial}{\partial A} \left[A^\dagger \frac{A}{V_R} \hat{\mathcal{P}} - A^\dagger \mathcal{P}(t) \right] \frac{dA}{dp} \right) dt \\
& - \int_{t_0}^{t_f} \left(\frac{\partial}{\partial p} \left[A^\dagger \frac{A}{V_R} \hat{\mathcal{P}} - A^\dagger \mathcal{P}(t) \right] \right) dt \\
& - \left[\frac{\partial}{\partial N} \left[N^\dagger N \right] \frac{dN}{dp} \right]_{t_0}^{t_f}. \tag{A.31}
\end{aligned}$$

Many of the above terms with partial derivatives equate to zero. They are

$$\begin{aligned}
0 &= \frac{\partial}{\partial \psi} \left[N \frac{dN^\dagger}{dt} \right] = \frac{\partial}{\partial A} \left[N \frac{dN^\dagger}{dt} \right] = \frac{\partial}{\partial p} \left[N \frac{dN^\dagger}{dt} \right] \\
&= \frac{\partial}{\partial A} \left\langle \psi^\dagger H \psi \right\rangle_{E,\mathcal{D},\Omega} = \frac{\partial}{\partial A} \left\langle \psi^\dagger G \psi \right\rangle_{E,\mathcal{D},\Omega} \\
&= \frac{\partial}{\partial \psi} \left[A^\dagger \mathcal{P}(t) \right] = \frac{\partial}{\partial N} \left[A^\dagger \mathcal{P}(t) \right] = \frac{\partial}{\partial A} \left[A^\dagger \mathcal{P}(t) \right]
\end{aligned}$$

Other partials can be evaluated directly. They are

$$\begin{aligned}
\frac{\partial}{\partial \psi} \left\langle \psi, H^\dagger \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} &= \left\langle H^\dagger \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} \\
\frac{\partial}{\partial \lambda} \lambda \left\langle \psi^\dagger, G \psi \right\rangle_{E,\mathcal{D},\Omega} &= \left\langle \psi^\dagger, G \psi \right\rangle_{E,\mathcal{D},\Omega} \\
\frac{\partial}{\partial \psi} \left\langle \psi, G^\dagger \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} &= \left\langle G^\dagger \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} \\
\frac{\partial}{\partial N} \left[N \frac{dN^\dagger}{dt} \right] &= \frac{dN^\dagger}{dt} \\
\frac{\partial}{\partial N} \left\langle Nb^\dagger N^\dagger \right\rangle_{E,\mathcal{D},\Omega} &= \left\langle b^\dagger N^\dagger \right\rangle_{E,\mathcal{D},\Omega} \\
\frac{\partial}{\partial \psi} \hat{\mathcal{P}} &= \left\langle \Sigma_E \right\rangle_{E,\mathcal{D},\Omega} \\
\frac{\partial}{\partial A} \left[A^\dagger \frac{A}{V_R} \hat{\mathcal{P}} \right] &= \frac{A^\dagger}{V_R} \hat{\mathcal{P}} \\
\frac{\partial}{\partial N} \left[N^\dagger N \right] &= N^\dagger
\end{aligned}$$

We now aggregate the terms that multiply $\frac{d\psi}{dp}$, $\frac{d\lambda}{dp}$, $\frac{dN}{dp}$, and $\frac{dA}{dp}$ to write

$$\begin{aligned}
\frac{d\mathcal{L}}{dp} = & \int_{t_0}^{t_f} \left\{ \left[\left\langle \frac{\partial R}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} \right. \right. \\
& + \left\langle N^\dagger \frac{\partial bN}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A^\dagger \frac{A}{V_R} \Sigma_E \right\rangle_{E,\mathcal{D},\Omega} \Big] \frac{d\psi}{dp} \Big\} dt \\
& + \int_{t_0}^{t_f} \left\langle \psi^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} \frac{d\lambda}{dp} dt \\
& + \int_{t_0}^{t_f} \left\{ \left[\left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right. \right. \\
& \quad \left. \left. + \frac{dN^\dagger}{dt} + \left\langle b^\dagger N^\dagger \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} \right] \frac{dN}{dp} \right\} dt \\
& + \int_{t_0}^{t_f} \left\{ \left[\left\langle \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle N^\dagger \frac{\partial bN}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} - \frac{A^\dagger}{V_R} \hat{\mathcal{P}} \right] \frac{dA}{dp} \right\} dt \\
& + \int_{t_0}^{t_f} \left\{ \left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right. \\
& \quad \left. + \left\langle N^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} + A^\dagger \frac{\partial \mathcal{P}(t)}{\partial p} \right\} dt \\
& - \left[N^\dagger \frac{dN}{dp} \right]_{t=t_f} + \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0}. \tag{A.32}
\end{aligned}$$

We are now ready to extract the adjoint equations. First, the expression that multiplies $\frac{dA}{dp}$ yields the following adjoint normalization equation:

$$A^\dagger = \frac{V_R}{\hat{\mathcal{P}}} \left\langle \frac{\partial R}{\partial A} + N^\dagger \frac{\partial bN}{\partial A} \right\rangle_{E,\mathcal{D},\Omega}.$$

Using Eq. (A.27) this can be written as

$$A^\dagger = \frac{V_R}{\hat{\mathcal{P}}} \left\langle \frac{\partial R}{\partial A} + N^\dagger \frac{\partial bN}{\partial \Psi} \psi \right\rangle_{E,\mathcal{D},\Omega}. \tag{A.33}$$

Next, the stationarity condition corresponding to $\frac{d\psi}{dp}$ requires that

$$H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger = S^\dagger$$

where

$$S^\dagger = \frac{\partial R}{\partial \psi} + N^\dagger \frac{\partial bN}{\partial \psi} - A^\dagger \frac{A}{V_R} \Sigma_E.$$

This expression requires some extra attention. We know that the operator $H^\dagger - \lambda G^\dagger$ is singular. The Fredholm alternative theorem[50] requires that for $H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger = S^\dagger$ to have a unique solution, the following must be true:

$$\langle S^\dagger, \psi \rangle_{E, \mathcal{D}, \Omega} = 0 \quad (\text{A.34})$$

where ψ satisfies the forward equation $H\psi - \lambda G\psi = 0$. Using Eq. (A.28), we find

$$\begin{aligned} 0 &= \langle S^\dagger, \psi \rangle_{E, \mathcal{D}, \Omega} \\ &= \left\langle \frac{\partial R}{\partial \psi} \psi \right\rangle_{E, \mathcal{D}, \Omega} + \left\langle AN^\dagger \frac{\partial bN}{\partial \Psi} \psi \right\rangle_{E, \mathcal{D}, \Omega} - \frac{A^\dagger}{V_R} \mathcal{P}(t). \end{aligned}$$

Now, if we multiply Eq. (A.33) by A , manipulate to solve for $AN^\dagger \frac{\partial bN}{\partial \Psi} \psi$, and substitute the result into the preceding equation, we find that Eq. (A.34) is satisfied if our QOI satisfies

$$0 = \left\langle \frac{\partial R}{\partial \psi} \psi \right\rangle_{E, \mathcal{D}, \Omega} - \left\langle A \frac{\partial R}{\partial A} \right\rangle_{E, \mathcal{D}, \Omega}. \quad (\text{A.35})$$

For most practical QOIs of interest, this will be satisfied. For example, any form of R that is bilinear in A and ψ will satisfy this relationship. The stationarity condition

corresponding to $\frac{d\lambda}{dp}$ corresponds to

$$\langle \psi^\dagger, G\psi \rangle_{E,\mathcal{D},\Omega} = 0$$

which says that the adjoint flux must be orthogonal to the fission source. Finally, the adjoint Bateman equation is

$$\begin{aligned} \frac{dN^\dagger}{dt} &= \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} - B^\dagger N^\dagger \\ &\quad - \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} + A^\dagger \frac{A}{V_R} \frac{\partial}{\partial N} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} \end{aligned}$$

If the stationarity conditions hold, then Eq. (A.32) reduces to

$$\begin{aligned} \frac{dQ}{dp} &= \int_{t_0}^{t_f} \left\{ \left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right. \\ &\quad \left. + \left\langle N^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - A^\dagger \frac{A}{V_R} \frac{\partial}{\partial p} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} + A^\dagger \frac{\partial \mathcal{P}(t)}{\partial p} \right\} dt \\ &\quad - \left[N^\dagger \frac{dN}{dp} \right]_{t=t_f} + \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0}. \end{aligned} \quad (\text{A.36})$$

Once again, the only terms left to deal with are $\frac{dN}{dp}$ evaluated at $t = t_0$ and $t = t_f$.

We eliminate the terminal term by imposing the terminal condition

$$N^\dagger(t_f) = 0.$$

The initial term, as discussed above, is assumed to be known from other data or serves as an avenue for computing sensitivities with respect to initial conditions. It remains in the sensitivity equation.

To review, the k-eigenvalue adjoint equations for a time-integrated QOI are

$$\begin{aligned} \frac{dN^\dagger}{dt} &= \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} - B^\dagger N^\dagger \\ &\quad - \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} + A^\dagger \frac{A}{V_R} \frac{\partial}{\partial N} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} \end{aligned} \quad (\text{A.37})$$

$$H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger = \frac{\partial R}{\partial \psi} + N^\dagger \frac{\partial bN}{\partial \psi} - A^\dagger \frac{A}{V_R} \Sigma_E \quad (\text{A.38})$$

$$\left\langle \psi^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} = 0 \quad (\text{A.39})$$

$$A^\dagger = \frac{V_R}{\hat{\mathcal{P}}} \left\langle \frac{\partial R}{\partial A} + N^\dagger \frac{\partial bN}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} \quad (\text{A.40})$$

with terminal condition

$$N^\dagger(t_f) = 0. \quad (\text{A.41})$$

If this system is satisfied, the sensitivity equation is

$$\begin{aligned} \frac{dQ}{dp} &= \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0} + \int_{t_0}^{t_f} \left\{ \left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right. \\ &\quad \left. + \left\langle N^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - A^\dagger \frac{A}{V_R} \frac{\partial}{\partial p} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} + A^\dagger \frac{\partial \mathcal{P}(t)}{\partial p} \right\} dt \end{aligned} \quad (\text{A.42})$$

Note: Eqs. (A.37)–(A.42) are identical to Eqs. (3.65)–(3.70).

We now consider the case of a terminal QOI, or a QOI that depends only on the solution at $t = t_f$:

$$Q_f = \left\langle R(N(t_f), \psi(t_f), \lambda(t_f), p, t_f) \right\rangle_{E,\mathcal{D},\Omega}. \quad (\text{A.43})$$

Note that this QOI may reasonably depend on the terminal eigenvalue, $k_{\text{eff}}(t_f)$. Therefore, we must account for this dependence. The derivation begins by noting

that

$$\frac{dQ_f}{dp} = \frac{d}{dt_f} \frac{dQ}{dp}.$$

Again, recall the following rule of calculus:

$$\frac{d}{dt_f} \int_{t_0}^{t_f} f(t, t_f) dt = f(t_f, t_f) + \int_{t_0}^{t_f} \frac{df}{dt_f} dt.$$

This rule is applied to take $\frac{d}{dt_f}$ of the appropriate version of Eq. (A.32). This version accounts for the dependence on $\lambda(t_f)$ and is written as follows:

$$\begin{aligned} \frac{d\mathcal{L}}{dp} = & \int_{t_0}^{t_f} \left\{ \left[\left\langle \frac{\partial R}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} \right. \right. \\ & + \left\langle N^\dagger \frac{\partial bN}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A^\dagger \frac{A}{V_R} \Sigma_E \right\rangle_{E,\mathcal{D},\Omega} \left. \right] \frac{d\psi}{dp} \right\} dt \\ & + \int_{t_0}^{t_f} \left(\left[\left\langle \frac{\partial R}{\partial \lambda} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \psi^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} \right] \frac{d\lambda}{dp} \right) dt \\ & + \int_{t_0}^{t_f} \left\{ \left[\left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right. \right. \\ & + \frac{dN^\dagger}{dt} + \left\langle b^\dagger N^\dagger \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} \left. \right] \frac{dN}{dp} \right\} dt \\ & + \int_{t_0}^{t_f} \left\{ \left[\left\langle \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle N^\dagger \frac{\partial bN}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} - \frac{A^\dagger}{V_R} \hat{\mathcal{P}} \right] \frac{dA}{dp} \right\} dt \\ & + \int_{t_0}^{t_f} \left\{ \left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right. \\ & + \left\langle N^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} + A^\dagger \frac{\partial \mathcal{P}(t)}{\partial p} \left. \right\} dt \\ & - \left[N^\dagger \frac{dN}{dp} \right]_{t=t_f} + \left[N^\dagger \frac{dN}{dp} \right]_{t=t_0}. \end{aligned} \tag{A.44}$$

According to our differentiation rule, after applying $\frac{d}{dt_f}$ to Eq. (A.44), some terms will be time integrated and others will be evaluated at $t = t_f$. Let us first examine

the resulting terms that are inside the integral. Define

$$\begin{aligned} N_f^\dagger &= \frac{\partial N^\dagger}{\partial t_f} \\ \psi_f^\dagger &= \frac{\partial \psi^\dagger}{\partial t_f} \\ A_f^\dagger &= \frac{\partial A^\dagger}{\partial t_f}. \end{aligned}$$

Then the terms inside the integral are

$$\begin{aligned} \int_{t_0}^{t_f} \frac{df(t, t_f)}{dt_f} dt &= \int_{t_0}^{t_f} \left\{ \left[- \left\langle H^\dagger \psi_f^\dagger - \lambda G^\dagger \psi_f^\dagger \right\rangle_{E, \mathcal{D}, \Omega} + \left\langle N_f^\dagger \frac{\partial bN}{\partial \psi} \right\rangle_{E, \mathcal{D}, \Omega} \right. \right. \\ &\quad \left. \left. - \left\langle A_f^\dagger \frac{A}{V_R} \Sigma_E \right\rangle_{E, \mathcal{D}, \Omega} \right] \frac{d\psi}{dp} \right\} dt \\ &\quad + \int_{t_0}^{t_f} \left\langle \psi_f^\dagger, G\psi \right\rangle_{E, \mathcal{D}, \Omega} \frac{d\lambda}{dp} dt \\ &\quad + \int_{t_0}^{t_f} \left\{ \left[- \left\langle \psi_f^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E, \mathcal{D}, \Omega} + \frac{dN_f^\dagger}{dt} + \left\langle b^\dagger N_f^\dagger \right\rangle_{E, \mathcal{D}, \Omega} \right. \right. \\ &\quad \left. \left. - \left\langle A_f^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial N} \right\rangle_{E, \mathcal{D}, \Omega} \right] \frac{dN}{dp} \right\} dt \\ &\quad + \int_{t_0}^{t_f} \left\{ \left[\left\langle N_f^\dagger \frac{\partial bN}{\partial A} \right\rangle_{E, \mathcal{D}, \Omega} - \frac{A_f^\dagger}{V_R} \hat{\mathcal{P}} \right] \frac{dA}{dp} \right\} dt \\ &\quad + \int_{t_0}^{t_f} \left\{ - \left\langle \psi_f^\dagger, \frac{\partial}{\partial p} [H\psi - \lambda G\psi] \right\rangle_{E, \mathcal{D}, \Omega} \right. \\ &\quad \left. + \left\langle N_f^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E, \mathcal{D}, \Omega} - \left\langle A_f^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial p} \right\rangle_{E, \mathcal{D}, \Omega} + A_f^\dagger \frac{\partial \mathcal{P}(t)}{\partial p} \right\} dt \end{aligned}$$

From this we obtain the expected adjoint equations:

$$\begin{aligned}
\frac{dN_f^\dagger}{dt} &= -B^\dagger N_f^\dagger + \left\langle \psi_f^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} + A_f^\dagger \frac{A}{V_R} \frac{\partial}{\partial N} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} \\
H^\dagger \psi_f^\dagger - \lambda G^\dagger \psi_f^\dagger &= N_f^\dagger \frac{\partial bN}{\partial \psi} - A_f^\dagger \frac{A}{V_R} \Sigma_E \\
\left\langle \psi_f^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} &= 0 \\
A_f^\dagger &= \frac{V_R}{\hat{\mathcal{P}}} \left\langle N_f^\dagger \frac{\partial bN}{\partial A} \right\rangle_{E,\mathcal{D},\Omega}
\end{aligned}$$

We now return to the $f(t_f, t_f)$ portion of Eq. (A.44) after $\frac{d}{dt_f}$ has been applied. The terms are

$$\begin{aligned}
f(t_f, t_f) &= \left[\left[\left\langle \frac{\partial R}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger \right\rangle_{E,\mathcal{D},\Omega} \right. \right. \\
&\quad \left. \left. + \left\langle N^\dagger \frac{\partial bN}{\partial \psi} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A^\dagger \frac{A}{V_R} \Sigma_E \right\rangle_{E,\mathcal{D},\Omega} \right] \frac{d\psi^\dagger}{dp} \right]_{t=t_f} \\
&\quad + \left[\left[\left\langle \frac{\partial R}{\partial \lambda} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \psi^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} \right] \frac{d\lambda}{dp} \right]_{t=t_f} \\
&\quad + \left[\left[\left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right. \right. \\
&\quad \left. \left. + \frac{dN^\dagger}{dt} + B^\dagger N^\dagger - \left\langle A^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} \right] \frac{dN}{dp} \right]_{t=t_f} \\
&\quad + \left[\left[\left\langle \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} + \left\langle N^\dagger \frac{\partial bN}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} - \frac{A^\dagger}{V_R} \hat{\mathcal{P}} \right] \frac{dA}{dp} \right]_{t=t_f} \\
&\quad + \left[\left[\left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right. \right. \\
&\quad \left. \left. + \left\langle N^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - A^\dagger \left[\frac{A}{V_R} \frac{\partial}{\partial p} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} + \frac{\partial \mathcal{P}(t)}{\partial p} \right] \right] \right]_{t=t_f}
\end{aligned} \tag{A.45}$$

The terminal condition for N_f^\dagger is extracted from Eq. (A.45). First, recall our terminal condition for N^\dagger , which still holds:

$$N^\dagger(t_f) = 0.$$

Taking the total derivative with respect to time, we have

$$N_f^\dagger(t_f) + \frac{dN^\dagger}{dt} \Big|_{t=t_f} = 0.$$

To eliminate $\frac{dN}{dp} \Big|_{t=t_f}$ from Eq. (A.45), we must have

$$\frac{dN^\dagger}{dt} \Big|_{t=t_f} = A^\dagger \frac{A}{V_R} \frac{\partial}{\partial N} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} + \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega}$$

Then, because $N_f^\dagger(t_f) = -\frac{dN^\dagger}{dt} \Big|_{t=t_f}$, our terminal condition is

$$N_f^\dagger(t_f) = \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - A^\dagger \frac{A}{V_R} \frac{\partial}{\partial N} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega}.$$

We have not yet defined $\psi(t_f)$ and $A^\dagger(t_f)$, however. They are made to eliminate $\frac{d\psi}{dp} \Big|_{t=t_f}$ and $\frac{dA}{dp} \Big|_{t=t_f}$, respectively, from Eq. (A.45). Accounting for $N^\dagger(t_f) = 0$, these expressions are

$$\begin{aligned} H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger &= \frac{\partial R}{\partial \psi} - A^\dagger \frac{A}{V_R} \Sigma_E \\ \left\langle \psi^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} &= - \left\langle \frac{\partial R}{\partial \lambda} \right\rangle_{E,\mathcal{D},\Omega} \\ A^\dagger &= \frac{V_R}{\hat{\mathcal{P}}} \left\langle \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} \end{aligned}$$

It is straightforward to show that (for almost all forms of R)

$$\left\langle \psi, \frac{\partial R}{\partial \psi} - A^\dagger \frac{A}{V_R} \Sigma_E \right\rangle_{E,\mathcal{D},\Omega} = 0$$

by substituting for A^\dagger ; therefore, the adjoint transport equation for the terminal condition has a unique solution.

In summary, for a terminal QOI, we have the following adjoint equations:

$$\frac{dN_f^\dagger}{dt} = -B^\dagger N_f^\dagger + \left\langle \psi_f^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} + A_f^\dagger \frac{A}{V_R} \frac{\partial}{\partial N} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} \quad (\text{A.46})$$

$$H^\dagger \psi_f^\dagger - \lambda G^\dagger \psi_f^\dagger = N_f^\dagger \frac{\partial bN}{\partial \psi} - A_f^\dagger \frac{A}{V_R} \Sigma_E \quad (\text{A.47})$$

$$\left\langle \psi_f^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} = 0 \quad (\text{A.48})$$

$$A_f^\dagger = \frac{V_R}{\hat{\mathcal{P}}} \left\langle N_f^\dagger \frac{\partial bN}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} \quad (\text{A.49})$$

The terminal condition, $N_f^\dagger(t_f)$, is

$$N_f^\dagger(t_f) = \left\langle \frac{\partial R}{\partial N} \right\rangle_{E,\mathcal{D},\Omega} - A^\dagger \frac{A}{V_R} \frac{\partial}{\partial N} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial N} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \quad (\text{A.50})$$

where ψ^\dagger and A^\dagger satisfy

$$H^\dagger \psi^\dagger - \lambda G^\dagger \psi^\dagger = \frac{\partial R}{\partial \psi} - A^\dagger \frac{A}{V_R} \Sigma_E \quad (\text{A.51})$$

$$\left\langle \psi^\dagger, G\psi \right\rangle_{E,\mathcal{D},\Omega} = - \left\langle \frac{\partial R}{\partial \lambda} \right\rangle_{E,\mathcal{D},\Omega} \quad (\text{A.52})$$

$$A^\dagger = \frac{V_R}{\hat{\mathcal{P}}} \left\langle \frac{\partial R}{\partial A} \right\rangle_{E,\mathcal{D},\Omega} \quad (\text{A.53})$$

The solution to the forward and adjoint equations can then be cross-correlated using

the following sensitivity expression:

$$\begin{aligned}
\frac{dQ_f}{dp} = & \left[N_f^\dagger \frac{dN}{dp} \right]_{t=t_0} + \left[\left\langle \frac{\partial R}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle \psi^\dagger, \frac{\partial}{\partial p} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right. \\
& \left. - A^\dagger \left[\frac{A}{V_R} \frac{\partial}{\partial p} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} + \frac{\partial \mathcal{P}(t)}{\partial p} \right] \right]_{t=t_f} \\
& + \int_{t_0}^{t_f} \left\{ - \left\langle \psi_f^\dagger, \frac{\partial}{\partial p} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} \right. \\
& \left. + \left\langle N_f^\dagger \frac{\partial bN}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} - \left\langle A_f^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial p} \right\rangle_{E,\mathcal{D},\Omega} + A_f^\dagger \frac{\partial \mathcal{P}(t)}{\partial p} \right\} dt \quad (\text{A.54})
\end{aligned}$$

Equations (A.46)–(A.54) are identical to Eqs. (3.75)–(3.85), which were developed using the abstraction contributed by this dissertation.

APPENDIX B

DETAILED DESCRIPTION OF THE DEPLETION, ADJOINT , AND DEPLETION PERTURBATION SOLVER IMPLEMENTATIONS IN PDT

This appendix serves as a user’s manual and as code documentation for the depletion, adjoint, and depletion perturbation solver implementations in PDT. We provide a high level overview of these solvers in Sec. 4; here we give details and provide example input decks so that a new user can use and understand the solvers from the code level.

We adopt some notation in this appendix, described as follows

- Words in **typewriter font** correspond to files, classes, functions, or members in the PDT code
- The name “active” vector corresponds to the data member in PDT that is actually used for a transport or depletion calculation. For example, the active densities are those that the code actually uses to compute cross sections before a transport calculation. The active fluxes are those that are being iteratively updated during the source iteration procedure. Opposite of active are storage vectors. These are vectors where previous iterates or snapshots of the forward solutions are being stored.

The first half of this appendix is a user’s manual. It details how a modeler runs PDT in depletion, adjoint, and depletion perturbation mode. Here we provide descriptions of all options available to the modeler and, when necessary, give a brief overview of the path taken through the code. The second half of the appendix is code documentation. We describe the math and logic behind those classes and methods

that are not immediately self explanatory.

B.1 User’s Manual

The Forward Depletion Solver

First we describe the forward depletion solver and the additional information required by the user in the input file.

Specifying component depletion information

The depletion solver requires the user to provide additional information about the components in the problem. Figure B.1 gives an example `<component_def>` block for a depletion problem. New fields are `<A_number.int>`, `<Z_number.int>`, `<decay_product_def>`, and `<fission_product_def>`, which define the mass number, proton number, and decay/fission children of the component (if any). In addition to the flags shown here, the option `<do_not_delete.str>` may be set to `true` if the user does not want the component’s density to change. This flag defaults to false if not specified.

```
<component_def>
    <id.str>pu238</id.str>
    <data_file.str>dataFiles/TWR_P238A.cx</data_file.str>
    <scat_order.int>0</scat_order.int>
    <Z_number.int>94</Z_number.int>
    <A_number.int>238</A_number.int>
    <decay_product_def>
        <decay_product.str>u234</decay_product.str>
        <branching_ratio.fp>1.0</branching_ratio.fp>
    </decay_product_def>
    <fission_product_def>
        <fission_yield.fp>1.0</fission_yield.fp>
        <fission_product.str>mpu38</fission_product.str>
    </fission_product_def>
</component_def>
```

Figure B.1: Example `<component_def>` block for depletion in PDT input file

The current implementation requires the user to manually provide decay parent/child pairs and fission parent/chid pairs. If included, the `<decay_product_def>` block and `<fission_product_def>` block require a valid child name (an error will be thrown if the child does not exist) and a branching ratio or fission yield.

The `BaseComponent` class now includes members `Anum`, `Znum` and `dontDepleteMe`, as well as functions for accessing these members. Each component also stores a vector of `ReactionEvent` and `DecayEvent` classes, described below, which contain information about the reactions that the particular component may undergo and which, if any, components it may transmute to after the reaction.

The depletion solver implementation requires the storage and use of cross sections that were not previously used for transport calculations. These include σ_{n2n} , $\sigma_{n,\gamma}$, etc. For many nuclides, the half-life and energy-per-fission will be required. These can be specified in the cross-section file with MT IDs 457 and 458, respectively.

Another complication may arise if the total absorption cross section, MT 27, is not input explicitly. If it is not given in the cross-section file, it is derived by subtracting the total, zero-th order scattering cross section from the total cross section. The user is notified in either case by the depletion info summary.

The `bp_info` block

Depletion mode is activated in PDT using the `<bp_info>` block in the `<common>` block of the input file. An example `<bp_info>` block for the OPERATOR_SPLIT mode is shown in Fig. B.2. The various available options are described below.

```

<bp_info>
    <converge_data>both</converge_data>
    <phi_pointwise_tolerance.fp>.001</phi_pointwise_tolerance.fp>
    <density_pointwise_tolerance.fp>0.001</density_pointwise_tolerance.fp>
    <max_fixed_point_its.int>5</max_fixed_point_its.int>
    <ts_scheme>OPERATOR_SPLIT</ts_scheme>
    <depletion_cycles.int>1</depletion_cycles.int>
    <cycle_length.fp>38880000</cycle_length.fp>
    <density_time_step.fp>21600</density_time_step.fp>
    <depletion_time_step.fp>3888000</depletion_time_step.fp>
    <reaction>fission</reaction>
    <reaction>n2n</reaction>
    <reaction>n_gamma</reaction>
    <reaction>decay</reaction>
</bp_info>

```

Figure B.2: Example `<bp_info>` input file block for OPERATOR_SPLIT mode

Specifying reactions to be tracked in the problem

The family of `DepletionEvent` classes, which are implemented in `BaseDepletionEvent.h`, encapsulate the dynamics of the reactions and decays that cause a nuclide to be depleted and possibly transmuted to another nuclides. The user specifies a list of reactions to be tracked in the given problem using the `<reaction>` keyword. Absorption is always on. Optional supported reactions are `fission`, `n_gamma`, `n_2n`, `n_proton`, `n_alpha`, and `decay`.

The `BaseProblem` constructor in `BaseProblem.cc` reads the list of reactions specified by the user. The reactions are collected in a vector of class type `SupportedReaction`, which stores skeleton information such as name, the mass number change, ΔA , the proton number change, ΔZ , and the branching ratio. For depletion reactions, the mass number change and proton number change are known and hard-coded. For decay and fission, the code relies on the user input in the `<component_def>` block.

After the vector of supported reactions is generated, the `BaseProblem` constructor reads each component's info (cross sections, mass & proton number, etc.) from the input file. After this, the code proceeds to match parent/child pairs. For each sup-

ported reaction (except for fission and decay) and each component, the code checks all other components to see if the ΔA and ΔZ are such that the two components form a pair. If so, a `ReactionEvent` class is stored by the parent component. This class contains the same information as `SupportedEvent` in addition to the `CX_MT_id` pointing to the correct cross section for the particular reaction. For fission and decay, the code checks the `<component_def>` block for user-defined child ID and branching ratio.

After the list of `ReactionEvents` and `DecayEvents` has been assembled for each component, the code checks to see if the necessary cross sections have been provided. If the cross section for a particular reaction has not been included, an warning is printed. Then, if the edit `print_depletion_connectivity` is ON, a brief summary of the depletion data for the component is printed to the standard output. An example of this summary is given in Fig. B.3.

```
Depletion data for component pu238:
  Global Component ID: 15
  Z-Number: 94
  A-Number: 238
  Decay Constant, lambda = 2.50622e-10s^-1.
  Recoverable energy per Fission: 197.37 MeV.
  Child component, mpu38, is formed via the fission reaction with branching ratio 1.
  Child component, pu239, is formed via the n_gamma reaction with branching ratio 1.
  Child component, u234, is formed via decay with branching ratio 1.
```

Figure B.3: Example component depletion summary

Options available for OPERATOR_SPLIT and RUNGE_KUTTA mode

The time marching scheme for integrating the forward depletion equation is specified using the `<ts_scheme>` tag in the `<bp_info>` block. Options are `OPERATOR_SPLIT` and `RUNGE_KUTTA`. The settings for the two schemes differ, and we explain these settings here. The parsing is executed in `ProblemInput.cc`, and options are stored in

a structure `BP_Info`, a member of structure `Kind_Info`, defined in `BPIInfo.h`.

Figure B.2 gives an example <`bp_info`> block for OPERATOR_SPLIT mode. For this scheme, the time domain is split into depletion cycles. Each cycle may contain a number of depletion time steps, and each depletion time step may contain a number of density time steps. The options for OPERATOR_SPLIT are described as follows:

- <`depletion_cycles.int`> – The number of depletion cycles
- <`cycle_length.fp`> – The length of a cycle, in seconds
- <`depletion_time_step.fp`> – The length of a “flux” time step, in seconds
- <`density_time_step.fp`> – The length of a “density” time step, in seconds
- <`converge_data`> – Specifies what to test for convergence at the end of each depletion time step. Options are `both`, `flux`, `density`, or `none`.
- <`phi_pointwise_tolerance.fp`> – the maximum relative iterative error allowed in phi for convergence
- <`density_pointwise_tolerance.fp`> – the maximum relative iterative error allowed in the densities for convergence
- <`max_fixed_point_its.int`> – maximum number of iterations to try per depletion time step
- <`density_rk_scheme`> – specify the *implicit* Runge-Kutta scheme to be used to integrate the Bateman equation. Only `IMPLICIT_EULER` is supported as an option at this time, although the code in `BatemanSolver->deplete_nuclides` is written to support any implicit Runge-Kutta scheme.

Figure B.4 gives an example <bp_info> block for RUNGE-KUTTA mode. In this mode, the time domain is broken into cycles, and each cycle may contain a number of time steps.

```
<bp_info>
  <ts_scheme>RUNGE_KUTTA</ts_scheme>
  <runge_kutta_scheme>MODIFIED_EULER</runge_kutta_scheme>
  <depletion_cycles.int>450</depletion_cycles.int>
  <cycle_length.fp>86400</cycle_length.fp>
  <depletion_time_step.fp>86400</depletion_time_step.fp>
  <reaction>fission</reaction>
  <reaction>n2n</reaction>
  <reaction>n_gamma</reaction>
  <reaction>decay</reaction>
</bp_info>
```

Figure B.4: Example <bp_info> input file block for RUNGE-KUTTA mode

The options for RUNGE-KUTTA mode are described as follows:

- <runge_kutta_scheme> – Choose the Runge-Kutta scheme to be used to integrate the forward equations. Options are EXPLICIT_EULER, MODIFIED_EULER, RK4, HEUN-EULER, and BOGACKI-SHAMPINE, the latter two being embedded rules to be used when we are producing an adjoint based error estimate. See appendix E for information about these schemes.
- <depletion_cycles.int> – the number of depletion cycles
- <cycle_length.fp> – the cycle length, in seconds
- <depletion_time_step.fp> – the depletion time step, in seconds
- <bateman_subcycles.int> – specify the number of Bateman subcycles per time-step. This option can only be different from 1 if the scheme is EXPLICIT_EULER

or `IMPLICIT_EULER`, as it will prevent convergence at rates higher than first order. (note: this option is not pictured above).

`BaseComponent` functionality supporting depletion

The `BaseComponent` class was updated with several members and member access functions to support the depletion routines. Many of these are self-explanatory, but one requires some explanation.

First, the function `computeRxProdRate`, which is only valid for neutronics components, is a function designed to compute depletion-related reaction rates. The arguments to this function are an index, i , and a scalar flux vector ϕ . The function returns a microscopic reaction rate. The math performed is simply

$$\mathcal{R} = \sum_g \phi_g \sigma_{i,g} \gamma_i,$$

where $\sigma_{i,g}$ is the microscopic cross section corresponding to the i^{th} `ReactionEvent` stored by this particular component and γ_i is the proper branching ratio. If the reaction is absorption, which is a special case depletion reaction, the microscopic reaction rate is multiplied by -1.

Specifying a QOI for the Problem

The QOI classes play an integral role in the adjoint problems because it appears in the source terms of the adjoint equations. At the time of this publication, the files that contain the QOI classes are

- `BaseQOI.h` – The base class implementation
- `TotalInventoryQOI.h` – A QOI that is the total number of mols of a component in a certain volume

- ReactionRateQOI.h – A QOI that is the total reaction rate for a certain component in a certain volume
- ReactivityQOI.h – The reactivity QOI, $\rho = \frac{k_{\text{eff}} -}{k_{\text{eff}}}$

The QOI class is created if the `<QOI_def>` tag is present inside the `<prototype>` block of the input deck. An example QOI definition block is shown in Fig. B.5.

```

<!-- Specify a QOI -->
<QOI_def>
<QOI_type.str>total_inventory</QOI_type.str>
<QOI_id.str>u235_ejected</QOI_id.str>
<keyComp.str>u235</keyComp.str>
<QOI_reg.dim_bounds>
    <QOI_reg.dim_bounds.dim.int>0</QOI_reg.dim_bounds.dim.int>
    <QOI_reg.dim_bounds.div_start.int>49</QOI_reg.dim_bounds.div_start.int>
    <QOI_reg.dim_bounds.div_end.int>50</QOI_reg.dim_bounds.div_end.int>
</QOI_reg.dim_bounds>
<QOI_reg.dim_bounds>
    <QOI_reg.dim_bounds.dim.int>1</QOI_reg.dim_bounds.dim.int>
    <QOI_reg.dim_bounds.div_start.int>0</QOI_reg.dim_bounds.div_start.int>
    <QOI_reg.dim_bounds.div_end.int>0</QOI_reg.dim_bounds.div_end.int>
</QOI_reg.dim_bounds>
</QOI_def>

```

Figure B.5: Example `<QOI_def>` block

Options for this block are as follows:

- `<QOI_type.str>` – Specifies the type of QOI. Options are `total_inventory`, either `detector_response` or `reaction_rate`, or `reactivity`.
- `<QOI_id.str>` – A string QOI ID. This must match `<adjoint_QOI_id.str>` in the `<adjoint_def>` block if it is an adjoint problem.
- `<keyComp>` – required for the `detector_response` and `total_inventory` QOIs. Specifies which component the total reaction rate or total inventory will be computed for. Error will be thrown if the component name does not match a valid component name for the problem.

- <QOI_reg.dim_bounds> – Method for specifying which volume of the problem the QOI applies to. Same syntax as the material region specification in input file.

The QOI is a member of `SweepProblem` and is accessible via a public function, `QOI()`. The proper QOI constructor is called in constructor of `SweepProblem.h`. The QOI class is templated over a grid type. The problem materials vector, energy group aggregation information, `Kind_info`, and problem handle are all made available to the QOI after its construction. **Note:** the current QOI implementation operates only on the $t = t_f$ solution; it is not integrated in time.

Steady-state adjoint calculations

The steady-state adjoint transport solver is a useful tool for computing the sensitivity of the given QOI with respect to parameters defined by the modeler. The problem that is solved is described in Sec. 4.3. Adjoint mode is activated by the inclusion of the `<adjoint_def>` block inside the `<common>` block. If the `adjoint_def` block is present but the `bp_info` block is not present, the code enters steady-state adjoint mode. An example `adjoint_def` block is given in Fig. B.6.

```

<problemID.str>tenCell_TWR_adjoint</problemID.str>
<adjoint_def>
    <adjoint_QOI_id.str>tenCell_HowMuchPu239_Ejected</adjoint_QOI_id.str>
    <forward_solution.str>STORE</forward_solution.str>
    <adjoint_QOI_do.str>COMPUTE_SENSITIVITY</adjoint_QOI_do.str>
    <orthogonalize_psi_dagger.str>true</orthogonalize_psi_dagger.str>
    <orthogonalize_S_dagger.str>true</orthogonalize_S_dagger.str>
    <param_vec_def>SIGMA_T</param_vec_def>
    <param_vec_def>SIGMA_F</param_vec_def>
    <param_vec_def>INITIAL_DENSITY</param_vec_def>
    <max_memory_MB.fp>100</max_memory_MB.fp>
    <max_checkpoint_period.int>100</max_checkpoint_period.int>
    <num_ckpt_dirs.str>SPECIFIED</num_ckpt_dirs.str>
    <num_ckpt_dirs.int>0</num_ckpt_dirs.int>
</adjoint_def>

```

Figure B.6: Example `<adjoint_def>` block

Many of the pictured tags are not active during steady state mode, but we provide the description here. The tags inside the `<adjoint_def>` block are

- `<problemID.str>` – This is a required tag for adjoint problems. Names of checkpointing files and subdirectories will reflect the problem ID. Note that `<problemID.str>` is a member of `<common>` and not `<adjoint_def>`.
- `adjoint_QOI_id.str` – The ID of the QOI to which the adjoint will correspond. This must match the ID of the QOI set by the `<QOI_def>` block, described below.
- `<forward_solution.location.str>` – Method for generating the forward solution. Options are `compute` or `files`. If the former, the forward solution will be computed at run time and the checkpointing scheme will be determined by the `<forward_solution.str>` tag. If the latter, the code will look for preexisting files that were generated during a previous run of the forward problem. This option is only active for adjoint-depletion problems.
- `<forward_solution.str>` – Method for checkpointing the forward solution.

Options are `STORE`, `STORE_MOMENTS`, `CHECKPOINT`, `CHECKPOINT_MOMENTS`, and `INTERPOLATE_MOMENTS`. In steady-state mode, only `STORE` and `CHECKPOINT` are supported.

- `<adjoint_QOI_do.str>` – What to do with the adjoint variable. Options are `NOTHING`, `COMPUTE_SENSITIVITY`, `COMPUTE_SENSITIVITY_AND_ERROR`, or `COMPUTE_ERROR`. The latter two options are only available for depletion problems. If `NOTHING` is specified, the adjoint solution will not be computed. This option may be useful for generating checkpointing files for use in later, separate adjoint runs.
- `<orthogonalize_psi_dagger.str>` – option of whether or not to orthogonalize ψ^\dagger to the forward fission source (that is, whether or not to enforce Eq. (3.67) or Eq. (3.77)). This is not a required tag. Options are `true` or `false`. Default is `true`.
- `<orthogonalize_S_dagger.str>` – option of whether or not to orthogonalize S^\dagger to ψ before and MSA_solve. This is not a required tag. Options are `true` or `false`. Default is `true`.
- `<param_vec_def>` – a listing of parameters to which the QOI sensitivity will be computed. This will be discussed in further detail below.
- `<max_memory_MB.fp>` – The maximum memory footprint in MB that will be allowed for checkpointing the forward solution. If we are in `STOR_ALL` or `STOR_MOM` mode and the maximum footprint will be exceeded, the code automatically switches to `CKPT_ALL` or `CKPT_MOM` mode, respectively. If the user selected `CKPT_ALL` or `CKPT_MOM` mode, the memory footprint will determine the checkpointing frequency (unless it is greater than

`<max_checkpoint_period.int>`). This is only active for depletion adjoint problems.

- `<max_checkpointing_period.int>` – specify the maximum number of cycles between writes to file. Applies to CKPT_ALL, CKPT_MOM, and INTP_MOM modes. The smaller of the periods between `<max_checkpoint_period.int>` and that resulting from `<max_memory_MB.fp>` wins. This is only active for depletion adjoint problems.
- `<num_ckpt_dirs.str>` – specify the number of directories to create for writing checkpoint files. This is to be used when running with multiple processors, as it is typically more efficient to have them write to different directories on the I/O system. Options are ONE_PER_CORE, SQRT_NUM_CORE, ALL_IN_HOME, and SPECIFIED. This is not a required tag. Default is ALL_IN_HOME.
- `<num_ckpt_dirs.int>` – Required if `<num_ckpt_dirs.str>` is SPECIFIED.
- `<terminal_density_source.str>` – Not required. Default is compute, which is the normal adjoint mode setting. Option `read_from_input` allows user to specify the terminal adjoint densities by writing them into the `<material_def>` block of the input file instead of using the typical QOI-related terminal adjoint density expressions developed in this dissertation. This option was used, for example, to facilitate “shuffling” during adjoint mode of the traveling wave reactor problem.

Those settings that are only active for depletion adjoint problems will be discussed below. For steady-state adjoint problems, we need to define the `<param_vec_def>` tag, which allows the user to specify which parameters to consider in the adjoint problem. Options are SIGMA_T, SIGMA_S, NU_SIGMA_F, SIGMA_F, NU, DECAY_CONSTANT,

and `INITIAL_DENSITY`. The option `ALL_DATA` will activate all parameters. If a parameter is active, it is active for every component. If the parameter does not exist for the particular component, the code produces zeros for sensitivity. For parameters that do not decay, however, a sensitivity is still produced for the `DECAY_CONSTANT`. Finally, the `INITIAL_DENSITY` option will produce sensitivity with respect to initial density (or, for steady-state problems, the given number density) of every component in every spatial cell.

If the code determines that we are in steady-state adjoint mode, the `SSAdjointSolver` class is instantiated and the method `ss_adjoint_solve()` is called. The problem may be either fixed-source or k-eigenvalue. The solver first allocates the proper storage for the forward solution and computes the total problem volume, which may be required for k-eigenvalue problems. The forward solution is computed using either a fixed-source solve or power iteration. If `STORE` mode, the full angular flux and density vector are stored in memory. If `CHECKPOINT` mode, the full angular flux and density vector are written to file, then immediately read back into RAM and stored.

The code then chooses the proper path for computing S^\dagger . This depends on the QOI and whether the problem is k-eigenvalue or fixed source. See Sec. 3.6 for the specific forms of S^\dagger , which is ultimately computed by the QOI class. Then, the code calls either the adjoint version of `ss_solve` or `msa_solve` to compute ψ^\dagger . The sensitivity equation is then evaluated by the QOI class for those parameters requested by the input file.

The sensitivity information is printed to the output file in a block called `QOI_SENSITIVITY`. The information is sorted by parameter type, component ID, and group number.

Depletion Perturbation (adjoint) calculations

To run a depletion perturbation problem (that is, a forward depletion solve followed by a backwards adjoint solve), specify both the `bp_info` and `adjoint_def` blocks inside the `common` block of the input file. A QOI must also be defined in the problem using the `QOI_def` block. Adjoint mode is only available using the `RUNGE_KUTTA` option for `ts_scheme` in the `bp_info` block.

An example `adjoint_def` block is pictured in Fig. B.6. For basic depletion perturbation problems, the major `adjoint_def` options to consider are the checkpoint period (`max_checkpoint_period.int`), maximum memory(`max_memory_MB.fp`), checkpoint scheme, (`forward_solution.str`), and the parameter vector definition (`param_vec_def`). The checkpointing schemes are outlined in Sec. 4.6; the other options are described above.

Other, more advanced options for the depletion perturbation solver are:

- `<forward_solution_location.str>` (default = `compute`) – Adjoint mode can be run without first computing the forward solution if the necessary info about the forward solution is already saved to checkpointing files. To activate this option, set this tag to `files`. This option is only available for the `CHECKPOINT`, `CHECKPOINT_MOMENTS`, and `INTERPOLATE_MOMENTS` checkpointing schemes. Note: the checkpointing scheme, problemID, and checkpointing frequency specified in the adjoint input deck must exactly match those that were used during the forward solve. This way, the proper forward solution information and file structure is present for the adjoint calculation.
- `<terminal_density_source>` (default = `compute`) – The terminal adjoint density calculation can be bypassed if this option is set to `read_from_input`. In this case, the terminal adjoint densities will be set to the density value given in

the material definition block of the input file. This was a useful option when running back-to-back adjoint calculations or when re-starting adjoint calculations.

If the code is in depletion perturbation mode, the `DplAdjointSolver` class is created and its method `dpl_adjoint_solve` is called. The first major calculation within this method is `compute_display_memStats`, a memory diagnostic routine that populates the data members in the `memoryStats` struct. The function `count_memory_load` loops over the unknowns in the problem and counts the total number of unknowns per snapshot of ψ , ϕ , and N . Then, depending on the checkpointing scheme, the RAM footprint per cycle is computed and ultimately the RAM footprint to store the entire problem. The code then checks the user-defined limits. If the limits will be exceeded, the code either changes the checkpointing scheme or changes the checkpointing frequency until the limits are satisfied.

After the memory diagnostics, the code enters one of two major blocks inside the `dpl_adjoint_solve` function. If the code is in `STORE` or `STORE_MOMENTS` mode, it enters the first block. If the checkpointing scheme is `CHECKPOINT`, `CHECKPOINT_MOMENTS`, or `INTERPOLATE_MOMENTS`, the code enters the second block. In any case, the general flow of the subroutine is the same.

First, the total memory to store the forward solution (or the discrete chunks of it) is re-computed, allocated, and printed to the output file for processing. The code then performs a loop over the forward cycles by repetitively calling `bp_solve`. If we are in one of the `STORE` modes, the forward solution (either ψ or the source moments) is stored to RAM by the `Checkpointer` class at each stage of each time step of each cycle. Otherwise, the forward solution is written to file at the intervals specified by the user (or corrected by the memory limitations).

The code then enters adjoint mode. The correct terminal condition is computed based on the problem type and QOI. The backwards time-march then begins. If we are in one of the `STORE` modes, the code proceed backwards without entering `RECOMPUTE` mode. Otherwise, the code alternates between `ADJOINT` mode and `RECOMPUTE` mode, recomputing the chunks of the forward solution as necessary, computing the adjoint solution, cross-correlating, and then throwing the data away.

The forward and adjoint solutions are cross-correlated at every stage. The `QOI` class handles the cross-correlation and the integration of the sensitivity equation. Likewise, it handles the integration of the error estimate. The forms of the inner product calculations that appear in the adjoint and sensitivity equations are discussed below.

Once the adjoint time-march has reached $t = t_0$ and the last bit of the sensitivity and/or error equations have been integrated, the QOI sensitivity is reported to the output file in a block called `QOI_SENSITIVITY`. The information is sorted by parameter type, component ID, and group number.

B.2 Code Documentation

The purpose of the remainder of this appendix is to document the logic and math behind some of the more complex and important methods and functions in the PDT implementation. We do not include documentation of simple or low-level functions and classes, as they are documented in the code. We focus more on the inner product and integral calculations that appear in the adjoint equations.

The `BatemanSolver` Class

The `BatemanSolver` class is implemented in `BatemanSolver.h`. The class is templated on the grid, so it has access to all data stored on the cells and elements. The class is instantiated in the constructor of the `BPSolver` class.

Major functions in the BatemanSolver class are

1. `compute_macroscopic_data` – this function computes the macroscopic cross-section data (Σ_t , Σ_f , $\nu\Sigma_f$, and E_f on each cell given an index into the `densityForward` vector. That is, it uses a checkpointed snapshot of the forward density solution to compute the macroscopic data that is used by the solver during sweeps and/or k-eigenvalue source updates.
2. `compute_bateman_matrix` – this function computes and returns B , the Bateman matrix, on the cell referenced by the cell-reference input. The matrix is computed by looping over each component in the cell, checking whether that component produces any other components in the cell by reactions or decays, and accumulating the production/destruction rates in the proper entries. This requires interaction with the `ReactionEvent` and `DecayEvent` classes belonging to each component. Note that absorption is always “on”, and is implemented as a special case.
3. `deplete_nuclides` – this is the work-horse depletion method for the `OPERATOR_SPLIT` mode. It is not called during `RUNGE_KUTTA` mode. Recall from Sec. 4.2.1 that the cell-averaged flux magnitude is treated as a linear function over the broad time step, and the Bateman equation is solved implicitly over the broad time steps, possibly using shorter time steps. Let $B(t)$ be the Bateman operator,

$$\frac{dN}{dt} = B(t)N,$$

where the time dependence comes from the linear time-dependence of the flux.

Our strategy is to decompose B as follows:

$$B(t) = DpR0 + \frac{t}{\Delta t} dR,$$

where $DpR0$ contains the initial decay + reaction rates and dR contains the change in the reaction rate over the time step Δt . This formulation prevents the need to recompute a Bateman matrix at each density time step.

The matrices are allocated and a loop over each component, `cmp`, occurs. For each component, a loop over its `ReactionEvents` and `DecayEvents` occurs. For each `ReactionEvent`, the code checks to make sure the child exists in this cell. If so, the index of that child is set to `c1Ind`. the component calls its `computeRxProdRate` method twice: first with the beginning-of-time-step flux, which gives an initial reaction rate for this reaction, \mathcal{R}_0 , and a second time with the flux change over the time step, which gives a delta reaction rate for this reaction, $\Delta\mathcal{R}$. Recall that \mathcal{R}_0 and $\Delta\mathcal{R}$ are microscopic reaction rates; that is, they have not been multiplied by the parent density. Then

$$DpR0[c1Ind][cmp] += \mathcal{R}_0,$$

$$dR[c1Ind][cmp] += \Delta\mathcal{R}.$$

Similarly for decay, we first subtract λ_{cmp} from $DpR0$:

$$DpR0[cmp][cmp] -= \lambda.$$

We then search for decay children in this cell. If they exist,

$$\text{DpR0}[\text{clInd}] [\text{cmp}] += \lambda\gamma,$$

where γ is the appropriate branching ratio.

Once these matrices are formed, the code enters a time march over the depletion time step using the (usually shorter) density time steps. A loop over density time steps progresses until the depletion time step length is reached. For stage s of density time step $t \in [t_n, t_{n+1}]$, the following linear system must be solved

$$[I - a_{ss}B_s]N_s = N_n + (t_{n+1} - t_n) \sum_{j=1}^{s-1} a_{sj}B_jN_j.$$

Here

$$B_j = \text{DpR0} + \frac{t_j}{t_{n+1} - t_n} \text{dR},$$

where $t_j = t_n + c_j$. The solution at $t = t_{n+1}$ is

$$N_{n+1} = N_n + (t_{n+1} - t_n) \sum_{i=1}^s b_i B_i N_i.$$

Recall that coefficients a , b , and c define the particular Runge-Kutta scheme. We solve the linear system using PDT's built-in LU solver with pivoting. Once the end of the depletion time step is reached, each cell updates its macroscopic cross sections with the new densities to prepare for the next transport solve.

4. `compute_stage_vector` – This method computes

$$N_s = N_n + \Delta t \sum_{i=1}^{s-1} a_{si} B_i N_i,$$

where the $B_i N_i$ s have been stored in `dpl_dNdt`.

5. `compute_and_store_dNdt` – This function computes the time-derivative of the densities given the active flux vector. It is used exclusively by explicit schemes in `RUNGE_KUTTA` mode. The function loops over cells. Within each cell, the Bateman matrix is computed using `compute_bateman_matrix` and a cell-averaged scalar flux. If we are not sub-cycling the Bateman equation, the code simply computes BN , stores it in `dpl_dNdt` on each cell, and exits. If we are sub-cycling the Bateman solution, the sub-cycle time step, δt , is computed. The sub-cycle proceeds as follows

$$N^{\ell+1} = N^\ell + \delta t BN^\ell.$$

The time derivative for stage s is accumulated (or smeared) as

$$(BN)_s = \frac{\sum_\ell \delta t BN^\ell}{\sum_\ell \delta t}.$$

6. `advance_implicit_RK` – the implicit version of `compute_and_store_dNdt`. The sub-cycle equation requires a linear solve, namely

$$[I - \delta t] N^{\ell+1} = N^\ell,$$

and the time derivative is smeared as above.

7. `set_adjoint_depletion_transport_source` – this function sets S^\dagger for source-driven problems. It takes a single argument: the index into the checkpointing vectors where the proper forward solution snapshot will be found. According

to Eq. (3.49),

$$S^\dagger = N^\dagger \frac{\partial bN}{\partial \psi}.$$

The adjoint densities are on the cells in the active vectors. Using Eq. (2.25), we find

$$\frac{\partial bN_i}{\partial \psi_{geq}} = \frac{1}{\sum_{e'} V_{e'}} \sum_j F_b^{cm} \left(\hat{\sigma}_{g,j \rightarrow i} N_j - \delta_{ij} \hat{\sigma}_{a,g,i} N_i \right).$$

The source vector is taken as a dot product over i , integrated over angle (because that's how PDT expects fixed sources), and placed in `q_fixed` on each element.

8. `compute_and_store_dNdtd_adjoint` – This function computes the time derivative of the adjoint densities (Eqs. (3.43), (3.48), (3.65), and (3.75)) for explicit methods. The time derivatives are written to the `dpl_dNdtd` vector on each cell.

This function takes the following arguments:

- (a) `dt` – time step
- (b) `stage` – stage index
- (c) `forwardSourceInd` – index of the forward solution vectors to reference
- (d) `ckpt_moments` – bool, true if we are in a checkpoint-moments mode
- (e) `isKeig` – true if k-eigenvalue problem
- (f) `keff` – k_{eff}
- (g) `A` – power normalization factor
- (h) `ratio` – $\frac{A^\dagger}{V_R}$

The function enters a loop over cells. For each cell, the Bateman matrix is

produced. Next, for each nuclide k , the variable f_k^\dagger is computed as

$$f_k^\dagger = \left\langle \psi^\dagger, \frac{\partial}{\partial N_k} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega} + \left\langle A^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial N_k} \right\rangle_{E,\mathcal{D},\Omega}.$$

These inner products are computed using the `compute_ip_psidppN` and `compute_ip_AdagRatSigmaE` functions, which are described below. The code then advances the adjoint densities through the time step, subcycling as described above if required, and the result is left in the `stage` index of `dpl_dNdt` on each cell.

9. `advance_implicit_RK_adjoint` – this function advances the adjoint densities for implicit methods. It is the implicit analog of `compute_and_store_dNdt_adjoint`.
10. `compute_ip_psidppN` – this function computes

$$\left\langle \psi^\dagger, \frac{\partial}{\partial N_k} [H\psi - \lambda G\psi] \right\rangle_{E,\mathcal{D},\Omega}$$

for nuclide k . The subroutine inputs are a cell reference, density vector source index, flux vector source index, nuclide index k , and a value for k_{eff} if it is a k-eigenvalue problem. This subroutine amounts to loops over the phase space variables to compute the inner product. The three main parts are the total collision part, scattering part, and the fission part. See below for notation definitions. The total collision term is

$$\left\langle \psi^\dagger, \frac{\partial}{\partial N_k} \Sigma_t \psi \right\rangle_{E,\mathcal{D},\Omega} = \sum_{e=1}^E \sum_{e'=1}^E \sum_{g=1}^G \sum_{q=1}^Q \psi_{e' \tilde{q} g}^\dagger \sigma_{t,k,g} M_{e,e'} \psi_{eqg} \omega_{q0}$$

The scattering term is

$$\left\langle \psi^\dagger, \frac{\partial}{\partial N_k} S\psi \right\rangle_{E,\mathcal{D},\Omega} = \sum_{e=1}^E \sum_{g=1}^G \sum_{q=1}^Q \sum_{g'=1}^G \sum_{m=1}^M \sum_{q'=1}^Q \sum_{e'=1}^E \psi_{eg\tilde{q}}^\dagger \omega_{q0} \sigma_{mq} \sigma_{s,k,\ell,g \rightarrow g'} M_{e,e'} \psi_{e'g'q'} \omega_{mq'}$$

Finally, the fission term is implemented assuming the number density spectrum for χ , which complicates the derivative with respect to a particular number density. Recall our representation for the neutron emission spectrum:

$$\chi_g = \frac{\sum_{k=1}^K N_k \chi_{g,k}}{\sum_{n=1}^K N_n},$$

where K is the number of fissionable nuclides present in the mixture. Given this, we ultimately need

$$\left\langle \psi^\dagger, \frac{\partial}{\partial N_k} F\psi \right\rangle_{E,\mathcal{D},\Omega} = \left\langle \psi^\dagger, \frac{\partial}{\partial N_k} \left[\frac{1}{4\pi} \frac{\sum_{k'=1}^K N_{k'} \chi_{k'}}{N_{k'}} \sum_{g'=1}^G \sum_{q'=1}^Q \nu \Sigma_{f,g'} \psi_{g',q'} \omega_{q'0} \right] \right\rangle_{E,\mathcal{D},\Omega}.$$

Define the following terms

$$\begin{aligned} \mathcal{P}_{1,e} &\equiv \sum_{k'=1}^K N_{k'} \\ \mathcal{P}_{2,eg} &\equiv \sum_{k'=1}^K N_{k'} \chi_{g,k'} \\ \mathcal{P}_{3,e} &\equiv \sum_{g'=1}^G \sum_{q'=1}^Q \nu_{k,g'} \sigma_{f,k,g'} \psi_{e,g',q'} \omega_{q'0} \\ \mathcal{P}_{4,e} &\equiv \sum_{g'=1}^G \sum_{q'=1}^Q \nu_{g'} \Sigma_{f,g'} \psi_{e,g',q'} \omega_{q'0} \end{aligned}$$

The discrete inner product in terms of these sums is

$$\left\langle \psi^\dagger, \frac{\partial}{\partial N_k} F\psi \right\rangle_{E,\mathcal{D},\Omega} = \frac{1}{k_{\text{eff}}} \sum_{e=1}^E \sum_{g=1}^G \sum_{q=1}^Q \sum_{e'=1}^E \omega_{q0} M_{e,e'} \psi_{e'g\tilde{q}}^\dagger \mathcal{M}_{q0} \left(\frac{\mathcal{P}_{3,e} \mathcal{P}_{2,e}}{\mathcal{P}_{1,e}} + \frac{\chi_{k,g} \mathcal{P}_{1,e} - \mathcal{P}_{2,e}}{\mathcal{P}_{2,e}^2} \mathcal{P}_{4,e} \right)$$

11. `compute_ip_AdagRatSigmaE` – This function computes

$$\left\langle A^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial N_k} \right\rangle_{E,\mathcal{D},\Omega}$$

for nuclide k in a particular cell. If the component does not have a fission cross section, this term is zero. Otherwise, the discrete sum in a particular cell is simply

$$\left\langle A^\dagger \frac{A}{V_R} \frac{\partial \Sigma_E \psi}{\partial N_k} \right\rangle_{E,\mathcal{D},\Omega} = \sum_{e=1}^E \sum_{g=1}^G \sum_{q=1}^Q \frac{A^\dagger A}{V_R} E_{f,k} \sigma_{f,k,g} \psi_{egq} V_e \omega_{q0}.$$

The function returns the sum over all cells.

12. `compute_density_residual` This function computes an estimate for the density residual (called $r_1^d(t)$ in Eq. (3.20)). The residual is formed as follows. The true Bateman solution satisfies

$$\frac{dN}{dt} - BN = 0.$$

The approximate solution satisfies

$$\frac{d\tilde{N}}{dt} - B\tilde{N} = r_1^d.$$

Thus

$$\begin{aligned}
r_1^d &= \frac{d\tilde{N}}{dt} - \frac{dN}{dt} - B\tilde{N} + BN \\
&= \frac{\tilde{N}_{n+1} - N_n}{\Delta t} - \frac{N_{n+1} - N_n}{\Delta t} - B\tilde{N} + BN \\
&= \frac{\tilde{N}_{n+1} - N_{n+1}}{\Delta t} - B\tilde{N} + BN
\end{aligned}$$

Before entering this routine, `BPSolver` has computed a low and high order estimate for N_{n+1} , the end-of-time-step Bateman solution and the corresponding flux solution. We treat the higher-order solution estimate as “exact”, or as N , and the calculation of the residual is straightforward.

13. `compute_Adagger` – this function performs the bulk of the calculation of A^\dagger , given by Eq. (3.78) as

$$A^\dagger = \frac{V_R}{\hat{\mathcal{P}}} \left\langle N^\dagger \frac{\partial bN}{\partial A} \right\rangle_{E,\mathcal{D},\Omega}.$$

This subroutine takes care of the inner product part of the calculation using the `ReactionEvent` class.

14. `compute_baseMSAsource` – this subroutine computes the right-hand-side of Eq. (3.76), which we call the base MSA source because it does not change at each MSA iteration. The term to be computed is

$$N^\dagger \frac{\partial bN}{\partial \psi} - \frac{A^\dagger A}{V_R} \Sigma_E,$$

which must be computed for each ψ , or for each element, group, and angle. The code takes care of the dot product portion, which is a straightforward differentiation of Eq. (2.25), and the calculation of Σ_E . Note that there is no inner product about these terms; we do, however, integrate the final result for

each element and group over the angular moments for storage as a source term.

The OPERATOR_SPLIT forward depletion implementation

The depletion routines are implemented in `BPSolver.h`. The `BPSolver` class inherits from `KSolver`, which inherits from `NeutronicsSolver`. The problem may be either source-driven or k-eigenvalue – the correct “transport solve” routine is called whenever necessary.

If we are in OPERATOR_SPLIT mode, each spatial cell allocates the following vectors:

- `bp_old_phi` – stores the beginning of depletion-time-step cell-average scalar flux (size = [nGroup])
- `bp_test_phi` – stores the latest guess for the end of depletion-time-step scalar flux (size = [nGroup])
- `densityCopy` – stores the beginning of depletion-time-step densities (size = [# components])
- `densityTest` – stores the latest guess for the end of depletion-time-step densities (size = [# components])

Pseudocode for the OPERATOR_SPLIT mode was given in Fig. 4.1 – here we elaborate on some of these steps. Before entering a loop over cycles, a transport solve is performed to get the initial flux. Then `bp_old_phi` and `densityCopy` are set to the beginning-of-run cell-average fluxes and densities, respectively. This copy is performed by the `BatemanSolver` class, implemented in `BatemanSolver.h`.

The code then enters a loop over the requested number of cycles. Within each cycle is a loop over time depletion time steps, where a counter `runningTime` is

accumulated until the cycle length is reached. Within each time step, the code performs these steps:

1. `BatemanSolver` copies the latest flux and density end-of-time-step guesses to `bp_test_phi` and `densityTest`, respectively. The initial guess is the beginning-of-time-step flux and density.
2. The function `deplete_nuclides` in `BatemanSolver` is called, which loops over each cell and performs the nuclide depletion over the time step (described below). After this call, the next guess for the end-of-time-step densities is in the active vector on each cell.
3. A transport solve is called to compute the next end-of-time-step flux guess using the densities computed in the previous step.
4. If requested, point-wise convergence is tested for the cell-averaged flux in each group and each cell (this compares the active scalar fluxes to `bp_test_phi`).
5. If requested, point-wise convergence is tested for each density on each cell (this compares the active densities to `densityTest`).
6. The code decides whether this time step has converged or if we have reached the maximum number of fixed point iterations for this time step

After a time step, `densityTest` and `bp_test_phi` are updated to the latest guesses for the end-of-time-step densities and fluxes to serve as the initial guess for the next time step. The code exits the solver after the loop over cycles has completed.

Now we give details of the `deplete_nuclides` function, which is a member of the `BatemanSolver` class. For each cell, this routine uses an implicit Runge-Kutta scheme to solve the Bateman equations using a linear guess for the time-dependence of the cell averaged scalar flux.

The RUNGE-KUTTA forward depletion implementation

Here we describe the depletion solve if the `<ts_scheme>` tag is set to RUNGE-KUTTA. If we are in this mode, the following vectors are allocated on each cell

- `densityCopy` – stores the beginning of depletion-time-step densities (size = [<# components])
- `dpl_dNdt` – stores the time derivative of each nuclide at each stage (size = [#stages][# components])

The code enters a loop over cycles. Within each cycle, there is a loop over depletion time steps until the cycle length is reached. Pseudocode for integration over a single time step was shown in Fig. 4.2.

In the case of an explicit Runge-Kutta scheme, the code proceeds as follows. For the first stage, the code calls the `compute_and_store_dNdt` method in the `BatemanSolver` class. This scheme simply computes BN , using the same steps described above, and stores the resulting vector in the first index of `dpl_dNdt`. This is repeated for each spatial cell.

If it is a multi-stage scheme, then the second stage vector is computed by calling the `compute_stage_vector` function of the `BatemanSolver` class. This method computes

$$N_s = N_n + \Delta t \sum_{i=1}^{s-1} a_{si} B_i N_i,$$

where the $B_i N_i$ s have been stored in `dpl_dNdt`. The loop then proceeds to the next stage, where it computes the flux stage vector. This stage vector is then used to compute the next $B_i N_i$, and the cycle continues.

After the stage vectors have been computed, the densities at the end of the time step are computing using another call to `compute_stage_vector`, except the

coefficients that are used are the b s:

$$N_{n+1} = N_n + \Delta t \sum_{i=1}^s b_i B_i N_i.$$

These densities are used to compute the end-of-time-step fluxes, and the code moved on to the next time step.

A first look at this code in `BPSolver.h` will reveal numerous blocks of code related to checkpointing and storing snapshots of the depletion solution. This code will be documented below. The code that is used for straightforward depletion problems using the Runge-Kutta scheme is only a small subset of what appears in the file.

The `AdjointController` structure

The `AdjointController` structure is a member of `Kind_info` and is implemented in `AdjointController.h`. It stores the options and settings requested by the user for an adjoint problem. An adjoint problem is specified with the `<adjoint_def>` block in the input file. An example adjoint definition block is given in Fig. B.6.

Parameters are defined by the `parameterDefinition` struct in `AdjointController.h`. They are classified by the `PARAM_DIMENSION` enum as either scalar, 1D, transfer, or initial condition. When the parameter is specified by the `<param_vec_def>` tag, the `active` flag in `parameterDefinition` is turned on. This will ultimately tell the QOI where to allocate space to store the parameter derivatives.

The adjoint transport solver

Here we describe the manner by which we solve an adjoint transport problem. We want to solve

$$H^\dagger \psi^\dagger = S^\dagger.$$

We are not worried about the form of S^\dagger here; just suppose it has been computed and it is sitting on `qfixed` on each element. If it's an adjoint k-eigenvalue problem, then the most recent fission source update is part of `qfixed`. If it's not a k-eigenvalue problem but there are fissionable materials, then we have lumped them into the scattering source.

We know that all we need to do is transpose the energy in-scatter source. First, we add a bool member to `Kind_Info` called `adjoint_mode`. If we are wanting to solve an adjoint transport problem, this is set to `true`; otherwise `false`. Second, we added adjoint inscattering and total source classes called `Neut_AdjointInscat` and `Neut_TotAdjointSource` in `SetInscat_Method.h` and `SetTotSource_Method.h`, respectively. These classes mimic `Neut_Inscat` and `Neut_TotSource` in their inheritance properties and structure – they simply transpose the matrix!

The adjoint source classes are instantiated inside the `get_inscat` and `get_totsource` methods of `NeutronicsSolver.h`. If the `adjoint_mode` tag is on, the solver is pointed to the transposed source methods. If the tag is off, the solver is pointed to the forward methods. The default setting for the tag is `false` at problem construction.

Inner product calculations in `BaseQOI.h`

We will now describe the major operations carried out by the members of `BaseQOI.h`. First is a series of inner product calculations of the form

$$\left\langle x^\dagger, \frac{\partial}{\partial p} Ax \right\rangle_{E,D,\Omega},$$

where x may be ψ or N and A is the transport, fission, Bateman, etc. operator. The parameter p belongs to a particular component (i.e. total cross section for group g in component i). The inner product requires a forward and adjoint solution. Thus, these functions take as inputs a global component index, `cmpInd`, and an index,

`srcInd`, pointing to the correct snapshot of the forward solution, ψ and N . `srcInd` may be split into a density source index and a flux source index if our checkpointing mode is such that we are only storing source moments.

A description of each inner product calculation follows. Assume that the correct snapshot of the forward solution has been found. Also, let k index the component to which the parameter corresponds. Finally, adopt the following notation:

- C = number of spatial cells
- E = number of elements on current cell
- Q = number of angles (for the particular group set)
- G = number of groups (in reality its a loop over groups per group set)
- M = degree of Legendre expansion (for this group set)
- K = number of components in current cell
- $M_{e,e'}$ = entry e, e' in cell mass matrix
- ω_{qm} = discrete-to-moment weight for angle q and moment m
- σ_{qm} = moment-to-discrete weight for angle q and moment m
- \tilde{q} – index of angle opposite of angle with index q
- `compute_dHdSigmaT` computes

$$\begin{aligned} \left\langle \psi^\dagger, \frac{\partial}{\partial \sigma_{k,t,g}} H \psi \right\rangle_{E,\mathcal{D},\Omega} &= \left\langle \psi_g^\dagger, N_k \psi_g \right\rangle_{E,\mathcal{D},\Omega} \\ &= \sum_{c=1}^C \sum_{e=1}^E \sum_{e'=1}^E \sum_{q=1}^Q \psi_{g,\tilde{q},e'}^\dagger \psi_{g,q,e} \omega_{q0} M_{e,e'} N_{c,k} \end{aligned}$$

for each group and stores the results in the `dHdSigmaT` vector, which is a member of `BaseQOI`.

- `compute_dHdSigmaS` computes

$$\left\langle \psi^\dagger, \frac{\partial}{\partial \sigma_{k,s,\ell,g' \rightarrow g}} H \psi \right\rangle_{E,\mathcal{D},\Omega}.$$

The discrete scattering source to angle q and group g on element e is

$$S_{g,q,e} = \sum_{m=0}^M \sigma_{q,m} \sum_{g'=1}^G \sum_{k=1}^K \sigma_{k,s,\ell,g' \rightarrow g} N_k \sum_{q'=1}^Q \omega_{q',m} \psi_{g',q',e}$$

where ℓ is the Legendre moment corresponding to moment m . Therefore

$$\begin{aligned} & \left\langle \psi^\dagger, \frac{\partial}{\partial \sigma_{k,s,\ell,g' \rightarrow g}} H \psi \right\rangle_{E,\mathcal{D},\Omega} \\ &= - \sum_{c=1}^C \sum_{e=1}^E \sum_{e'=1}^E \sum_{q=1}^Q \sum_{m=0}^M \delta_{\ell,m} \omega_{q,0} M_{e,e'} \psi_{g,\tilde{q},e'}^\dagger \sigma_{q,m} N_k \sum_{q'=1}^Q \omega_{q',m} \psi_{g',q',e}, \end{aligned}$$

which is stored for each g, g', ℓ in `dHdSigmaS`.

- `compute_dGdNuSigmaF` computes

$$\begin{aligned} & \left\langle \psi^\dagger, \frac{\partial}{\partial \nu \sigma_{k,f,g'}} G \psi \right\rangle_{E,\mathcal{D},\Omega} = \left\langle \psi_g^\dagger, \sigma_{q0} \chi_g N_k \phi_{g'} \right\rangle_{E,\mathcal{D},\Omega} \\ &= \sum_{c=1}^C \sum_{e=1}^E \sum_{e'=1}^E \sum_{g=1}^G \sum_{q=1}^Q \omega_{q,0} M_{e,e'} \sigma_{q0} \psi_{g,\tilde{q},e'}^\dagger N_k \chi_g \sum_{q'=1}^Q \psi_{g',q',e} \omega_{q',0} \end{aligned}$$

for each g' and stores the result in `dGdNuSigmaF`.

- `compute_dGdSigmaF` computes

$$\begin{aligned} \left\langle \psi^\dagger, \frac{\partial}{\partial \sigma_{k,f,g'}} G\psi \right\rangle_{E,\mathcal{D},\Omega} &= \left\langle \psi_g^\dagger, \omega_{q0} \chi_g N_k \phi_{g'} \nu_{k,g'} \right\rangle_{E,\mathcal{D},\Omega} \\ &= \sum_{c=1}^C \sum_{e=1}^E \sum_{e'=1}^E \sum_{g=1}^G \sum_{q=1}^Q \omega_{q,0} M_{e,e'} \omega_{q0} \psi_{g,\tilde{q},e'}^\dagger N_k \chi_g \nu_{k,g'} \sum_{q'=1}^Q \psi_{g',q',e} \omega_{q'0} \end{aligned}$$

for each g' and stores the result in `dGdSigmaF`.

- `compute_dGdNuBar` computes

$$\begin{aligned} \left\langle \psi^\dagger, \frac{\partial}{\partial \nu_{k,f,g'}} G\psi \right\rangle_{E,\mathcal{D},\Omega} &= \left\langle \psi_g^\dagger, \omega_{q0} \chi_g N_k \phi_{g'} \sigma_{k,f,g'} \right\rangle_{E,\mathcal{D},\Omega} \\ &= \sum_{c=1}^C \sum_{e=1}^E \sum_{e'=1}^E \sum_{g=1}^G \sum_{q=1}^Q \omega_{q,0} M_{e,e'} \omega_{q0} \psi_{g,\tilde{q},e'}^\dagger N_k \chi_g \sigma_{k,f,g'} \sum_{q'=1}^Q \psi_{g',q',e} \omega_{q'0} \end{aligned}$$

for each g' and stores the result in `dGdNuBar`.

- `compute_dPhatdSigmaF` computes

$$\frac{A^\dagger A}{V} \frac{\partial}{\partial \sigma_{k,f,g'}} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} = \frac{A^\dagger A}{V} \left\langle E_{f,k} N_k \psi_{g'} \right\rangle_{E,\mathcal{D},\Omega},$$

where E_f is the energy released per fission in component k . This inner product is

$$\frac{A^\dagger A}{V} \frac{\partial}{\partial \sigma_{k,f,g'}} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} = \frac{A^\dagger A}{V} \sum_{c=1}^C \sum_{e=1}^E V_e \sum_{q=1}^Q \omega_{q0} \psi_{g',q,e} N_k E_{f,k}$$

for each g' and stores the result in `dPhatdSigmaF`.

- `compute_dBNdSigmaA` computes

$$\left\langle N^\dagger, \frac{\partial bN}{\partial \sigma_{k,a,g}} \right\rangle_{E,\mathcal{D},\Omega} = N^\dagger \cdot \frac{\partial BN}{\partial \sigma_{k,a,g}} = N^\dagger \cdot \frac{\partial}{\partial \sigma_{k,a,g}} \left[- \sum_{g=1}^G \sum_{k=1}^K A \sigma_{k,a,g} \phi_g N_k F_b^{cm} \right]$$

where A is the flux normalization constant (it is set to $A = 1.0$ for source-driven problems), and ϕ_g is the cell averaged flux in group g , namely

$$\phi_g = \frac{\sum_e \phi_{g,e} V_e}{\sum_e V_e} = \frac{\sum_e \sum_q \psi_{e,g,q} \omega_{q,0} V_e}{V_c}.$$

Thus

$$\frac{\partial}{\partial \sigma_{k,a,g}} BN = -AN_k F_b^{cm} \frac{\sum_e \sum_q \psi_{e,g,q} \omega_{q,0} V_e}{V_c}.$$

and

$$\left\langle N^\dagger, \frac{\partial bN}{\partial \sigma_{k,a,g}} \right\rangle_{E,\mathcal{D},\Omega} = -N_k^\dagger A N_k F_b^{cm} \frac{\sum_e \sum_q \psi_{e,g,q} \omega_{q,0} V_e}{V_c}.$$

This is computed for each g and stored in `dBNdSigmaA`. Later we will simply note that $\frac{\partial BN}{\partial \sigma_t} = \frac{\partial BN}{\partial \sigma_a}$. Also, in the case that σ_a is derived at run-time by subtracting σ_s from σ_t , we find $\frac{\partial BN}{\partial \sigma_s} = -\frac{\partial BN}{\partial \sigma_a}$ for zero-th order moments.

- `compute_dBNdSigmaF` computes

$$\left\langle N^\dagger, \frac{\partial bN}{\partial \sigma_{k,f,g}} \right\rangle_{E,\mathcal{D},\Omega} = N^\dagger \cdot \frac{\partial BN}{\partial \sigma_{k,f,g}}$$

The term $\frac{\partial BN}{\partial \sigma_{k,f,g}}$ is non-zero for $\frac{dN_j}{dt}$ if nuclide k produced nuclide j via fission.

The code checks to see if a fission child of nuclide k exists in the cell. Define

$$\delta_{kj} = \begin{cases} 1 & \text{Nuclide } k \text{ produces nuclide } j \text{ via fission} \\ 0 & \text{else} \end{cases}$$

Then

$$\left\langle N^\dagger, \frac{\partial bN}{\partial \sigma_{k,f,g}} \right\rangle_{E,\mathcal{D},\Omega} = -\delta_{kj} N_j^\dagger A N_k \gamma_{kj} F_b^{\text{cm}} \frac{\sum_e \sum_q \psi_{e,g,q} \omega_{q,0} V_e}{V_c},$$

where γ_{kj} is the fission yield for nuclide j from nuclide k . Again, this is computed for every group g and the result is stored in `dBNdSigmaF`.

- `compute_dBNdDecayConstant` computes

$$\left\langle N^\dagger, \frac{\partial bN}{\partial \lambda_k} \right\rangle_{E,\mathcal{D},\Omega} = N^\dagger \cdot \frac{\partial BN}{\partial \lambda_k}.$$

Let δ_{kj} carry the same meaning as above, but let $\delta_{kk} = 1$ and $\gamma_{kk} = -1$. Then

$$\left\langle N^\dagger, \frac{\partial bN}{\partial \lambda_k} \right\rangle_{E,\mathcal{D},\Omega} = \delta_{kj} N_j^\dagger \gamma_{kj} N_k.$$

- `compute_dPhatdN` computes

$$\frac{\partial \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega}}{\partial N_k}$$

for the k^{th} nuclide in a particular cell. This term appears in Eq. (3.84), the terminal condition for the adjoint densities in k-eigenvalue problems. If the nuclide contributes to the power production, the inner product is

$$\frac{\partial}{\partial N_k} \left\langle \Sigma_E \psi \right\rangle_{E,\mathcal{D},\Omega} = \sum_{g=1}^G \sum_{e=1}^E \sum_{q=1}^{Q_e} \omega_{q,0} V_e \psi_{egq} E_{f,k} \sigma_{f,k,g}.$$

Otherwise it is zero.

- `compute_dGpsidN` and `compute_dHpsidN` compute the inner products

$$\left\langle \psi^\dagger, \frac{\partial}{\partial N_k} G\psi \right\rangle_{E,\mathcal{D},\Omega}$$

and

$$\left\langle \psi^\dagger, \frac{\partial}{\partial N_k} H\psi \right\rangle_{E,\mathcal{D},\Omega}$$

for the k^{th} nuclide in a particular spatial cell. The form of these calculations was covered in the discussion of the `BatemanSolver` class above. These terms appear in Eq. (3.84), the terminal condition for the adjoint densities in k-eigenvalue problems.

Sensitivity equation integration in `BaseQOI.h`

The `BaseQOI` class contains a member that integrates the sensitivity equation after each time step. Our technique for performing this integration was introduced in Sec. 3.5. Suppose we are only using a single time step, Δt , $t \in [a, b]$, and let us use the following model sensitivity expression to illustrate this process:

$$S = S_T - \int_a^b (\alpha t + \beta) dt,$$

where S_T represents terms evaluated at either $t = a$ or $t = b$ and the integrand represents cross-correlation terms. The sensitivity obtained by analytic integration is

$$S = S_T - \left[\frac{\alpha}{2} (b^2 - a^2) + \beta(b - a) \right].$$

Recall that we define a dummy variable z which satisfies

$$\frac{dz}{dt} = \alpha t + \beta$$

$$z(b) = S_T,$$

and that after integration backwards, $S = z(a)$. To see this, consider integrating $z(t)$ using an s stage Runge-Kutta rule:

$$\frac{z(b) - z(a)}{\Delta t} = \sum_{i=1}^s \alpha_i t_i + \beta_i.$$

Here α_i and β_i represent the cross-correlation terms in the sensitivity equation evaluated at stage i in time. The integration rule becomes

$$z(a) = z(b) - (b - a) \sum_{i=1}^s \alpha_i t_i + \beta_i.$$

For example, the second-order modified Euler scheme applied to this system is

$$S = z(a) = z(b) - (b - a) \frac{1}{2} [\alpha_a a + \beta_a + \alpha_b b + \beta_b].$$

If α and β are constant in time, we recover the exact sensitivity expression derived above:

$$S = S_T - \left[\frac{\alpha}{2} (b^2 - a^2) + \beta (b - a) \right].$$

The base and derived QOI classes work in tandem to perform this integration. Three vectors belonging to `BaseQOI` are allocated at problem outset:

- `dCCSCALARdt` – size: [# parameters][# components][# stages]
- `dCCONEDdt` – size[# parameters][# components][# groups][# stages]

- `dCCTRANSFERdt` – size[`# parameters`][`# components`][`# groups`][`# moments`][`# groups`][`# stages`]

During each time step of the backwards solve, the code computes each term in the sensitivity equation (for each component and each parameter) at each Runge-Kutta stage. These calculations require the inner product functions that are described above. The results of these calculations are stored in the proper dCC vector (depending on the parameter dimensionality).

At the end of the time step, the depletion-adjoint solver calls `integrate_cross_correlation`, which computes

$$\frac{dQ}{dp} \leftarrow \frac{dQ}{dp} - \Delta t \sum_{i=1}^s F_i^{(p)}.$$

Members of the derived QOI classes

At the time of this dissertation, three QOI classes are available in PDT. Here we describe some of their functionality. First, each QOI inherits a virtual function `compute_QOI`. This function computes

- `TotalInventoryQOI`:

$$Q = \sum_{c=1}^C \delta_c V_c N_{k,c} N_A,$$

where $\delta_c = 1.0$ if the cell is in the specified QOI sub-volume, else 0.0, V_c is the cell volume, subscript k indicates the component of interest, and N_A is Avogadro's number.

- `ReactionRateQOI`:

$$Q = \sum_{c=1}^C \delta_c \sum_{e=1}^E \sum_{g=1}^G \sum_{q=1}^Q V_e N_{k,c} \sigma_{t,k,g} \psi_{e,g,q} \omega_{q,0}.$$

- ReactivityQOI:

$$Q = \frac{k_{\text{eff}} - 1}{k_{\text{eff}}}.$$

Each class also has the function `compute_dCCdt`, which computes the value of the cross-correlation terms in the sensitivity expression at the current stage. This function is not in `BaseQOI` because it can be different for each QOI if the QOI has a time-integrated component. This function loops over each supported parameter, checks to see if it is `active`, computes the proper inner product, and stores the cross-correlation term in either `dCCSCALARdt`, `dCCONEDdt`, or `DCCTRANSFERdt`. For example, the procedure for parameter $\sigma_{t,g}$ in component k in stage s is

1. `compute_dHdSigmaT`
2. `compute_dBNdSigmaA`
3. For each group, g , $\text{dCCONEDdt}[\text{SIGMA_T}][k][g][s] = \text{dHdSigmaT}[g] - \text{dBNdSigmaA}[g]$

A similar inner-product and storage step is required for each parameter at each stage.

The term $\left\langle \frac{\partial R}{\partial \sigma_{t,k,g}} \right\rangle_{E,\mathcal{D},\Omega}$ is non-zero for the reaction rate QOI. The function `add_dQOI_dSigmaT` in `ReactionRateQOI.h` computes this term as

$$\begin{aligned} \left\langle \frac{\partial R}{\partial \sigma_{t,k,g}} \right\rangle_{E,\mathcal{D},\Omega} &= \left\langle \frac{\partial}{\partial \sigma_{t,k,g}} \psi \sum_{j=1}^K \sigma_{t,j} N_j \right\rangle_{E,\mathcal{D},\Omega} \\ &= \sum_{c=1}^C \sum_{e=1}^E \sum_q \psi_{e,g,q} \omega_{q,0} N_k V_e \end{aligned}$$

where the loop over cells only covers those that are in the QOI region.

The functions `set_terminal_condition_qfixed` and `set_tc_baseMSAsource` are used to compute the terminal condition source terms (right hand sides of Eq. (3.50) and Eq. (3.83), respectively. Similarly, the function `compute_hatAdagger` computes

\hat{A}^\dagger , if it exists. These functions are only required if $\frac{\partial R}{\partial \psi} \neq 0$, and the calculations are straightforward and similar to many of the inner products described above.

The `SSAdjointSolver` class

The `SSAdjointSolver` class is implemented in `SSAdjointSolver.h`. This solver is called if the input file specifies an adjoint problem but does not specify depletion data. The driver function is `ss_adjoint_solve()`. The flow of the function is fairly straightforward and heavily documented in the source code, thus we do not cover it in detail here. We note that it is broken into two blocks, one for `STORE` mode and one for `CHECKPOINT` mode. First the forward problem is solved, and subsequently the adjoint problem is solved. The form of the adjoint solution depends on whether the problem is a k-eigenvalue problem or not.

If it is a fixed-source problem, the function `perform_adjoint_fxdS_solve()` is called to compute the adjoint fluxes. This subroutine first tells the QOI class to initialize the sensitivity vectors. Then, the proper adjoint source is computed and stored in `qfixed`. Finally, the steady state solver is called with adjoint mode set to true, and the adjoint fluxes are computed.

The k-eigenvalue case is a bit more complicated. The subroutine `perform_adjoint_keig_solve()` handles this case. First, the temporary variable \hat{A}^\dagger is computed (see Eq. (3.82) for an example). Then, the homogeneous adjoint flux is solved using an adjoint k-eigenvalue solve. If the QOI is `REACTIVITY`, then we are done (there is no MSA Solve required). Otherwise, the MSA source is computed and an MSA solve is performed using the homogeneous ψ^\dagger for orthogonalization.

The `DplAdjointSolver` class

The `DplAdjointSolver` class is implemented in `DplAdjointSolver.h`. This solver is called if the input file specifies that the problem is both an adjoint problem

and a depletion problem. The driver subroutine is `dpl_adjoint_sollve()`, which is heavily documented in the source code and very procedural in nature, meaning that it is simply a flow of logic depending on the user specifications. Therefore, we do not document this subroutine here, as it would be much less effective than examining the source code.

Other subroutines do require some notes. The subroutine `adjoint_depletion_step()` integrates the adjoint variables backwards over a single cycle. The subroutine first compute the end-of-cycle adjoint fluxes. It then enters a loop over time-steps and, within each time step, a loop over stages. Within a stage, the code proceeds as follows

1. if (not first stage), compute ψ^\dagger
2. advance the index of the forward solution backwards one
3. recover forward fluxes (depends on checkpointing scheme, see `recover_forward_psis()` documentation)
4. cross-correlate
5. advance N^\dagger s according to Runge-Kutta rule

This procedure is exactly like the Runge-Kutta integration of the forward equations, but with the added complication of keeping track of the forward solution index.

Two subroutines exist to prepare for adjoint flux solves. The first, `prepare_for_adjoint_ss_solve()`, computes the adjoint fixed source and stores it in `qfixed` prior to a fixed-source adjoint solve. This calculation involves the right-hand-side of Eq. (3.49). This subroutine also updates Σ_t in each cell using the proper forward density solutions.

The second subroutine is for k-eigenvalue problems; it is

`prepare_for_adjoint_keig_solve()`. This subroutine does three major calculations: (a) solves for the current A^\dagger ; (b) solves for the homogeneous adjoint fundamental mode flux shape, which is required for orthogonalization of the adjoint flux solution; and (c) it computes and stores the “base MSA source”, or right-hand-side of Eq. (3.76) by calling the `compute_baseMSAsource` function in the `BatemanSolver` class.

The subroutine `initialize_sensitivity_vectors()` is called just before the adjoint solve commences. The logic here is somewhat convoluted because of the many forms of the sensitivity equation for the different QOIs. The overarching functionality here is to compute the terms in the sensitivity equation (either Eq. (3.52) is source-driven or Eq. (3.85) if k-eigenvalue) that are to be evaluated at $t = t_f$. First, if the QOI is `REACTIVITY`, everything is treated different (see Sec. 3.6.3). Here we calculate and normalize $\hat{\psi}^\dagger$, which is simply the homogeneous adjoint flux. The QOI class then uses this to calculate the terminal sensitivity terms.

For other QOIs, if the QOI does not depend on the flux, then $\hat{\psi}^\dagger$ and \hat{A}^\dagger , if it exists, are zero, and this sub-routine simply allocates the sensitivity vectors as necessary. On the other hand, if the QOI does depend on the flux (that is, $\frac{\partial R}{\partial \psi} \neq 0$), $\hat{\psi}^\dagger$ and \hat{A}^\dagger (if necessary) must be computed. In the k-eigenvalue case, this requires the homogenous adjoint k-eigenvalue solve, calculation and storage of the terminal condition MSA source (right hand side of Eq. (3.83)), which is done by the QOI class, and finally and MSA solve for $\hat{\psi}^\dagger$. Otherwise, if it is a source-driven problem, the right terminal adjoint fixed source (right hand side of Eq. (3.50)) must be calculated, again by the QOI class, and a fixed source solve is performed.

At the end of this calculation, $\hat{\psi}^\dagger$ is sitting on the active elements (if necessary) and is used by the QOI class in a subsequent call to compute the sensitivity terms

that are evaluated at $t = t_f$ as well as the terminal condition for N^\dagger .

The final subroutine in the `DplAdjointSolver` class that we describe is `recover_forward_psis()`. This subroutine is called during the `adjoint_depletion_step` subroutine to recover the forward angular fluxes at the current stage in order to form the adjoint source and operator terms. The manner by which the adjoint fluxes are recovered depends on the checkpointing scheme. If the scheme is `STORE` or `CHECKPOINT`, the angular fluxes are already in RAM and nothing is to be done. If the scheme is `STORE_MOMENTS` or `CHECKPOINT_MOMENTS`, the angular flux must be recovered from the stored source moments. This calculation is handled by the `compute_forward_psis` subroutine in the `BPSolver` class.

This subroutine becomes more complicated if the checkpointing scheme is the `INTERPOLATE_MOMENTS` scheme. First, if we are using this scheme and we have reached the end of the cycle, we do not need to perform any interpolation because the source moments are already stored. This is handled by a call to `compute_forward_psis` with a pointer to the first (zero-th) moment storage location.

If we are somewhere in the middle of the cycle, we must interpolate source moments between the beginning-of-cycle and end-of-cycle, which are stored in RAM. First, the interpolation is performed linearly using the known forward source index. This interpolation is performed by the `Checkpointer` class. Once the source moments have been calculated, a single sweep is performed to recover a guess for ϕ to form the initial flux guess, and then a full `ss_solve` is performed to converge the scalar flux.

APPENDIX C

LISTING OF OTHER VERIFICATION PROBLEMS

C.1 Infinite Medium, Pure Absorber Source-Driven Problem

We can derive analytic solutions for the depletion adjoint equations in the limit of a pure absorber in an infinite medium. For example, in this limit with two energy groups and a single pure absorbing material, the governing equations are

$$\frac{dN}{dt} + N(\phi_1 \hat{\sigma}_1 + \phi_2 \hat{\sigma}_2) = 0$$

$$N \hat{\sigma}_1 \psi_{e1q} = \frac{S_1}{4\pi}$$

$$N \hat{\sigma}_2 \psi_{e2q} = \frac{S_2}{4\pi}$$

$$t \in [0, T]$$

$$N(t=0) = N_0$$

where e and q are element and angle indices, respectively, the element scalar flux is $\phi_{eg} = \sum_q \psi_{egq} \omega_{q \rightarrow 0}$, and the cell-averaged scalar flux is $\phi_g = \frac{1}{V} \sum_e \phi_{eg} V_e$, where V_e and \mathbf{V} are the element and cell volumes, respectively (note: we model an infinite medium in PDT using a single, *really big* cell, so these volumes make sense for the discrete problem). Our task is to compute some QOI, a functional of the solution to these equations, and the sensitivity of this QOI with respect to the microscopic cross sections.

Consider solving these equations over a single time step T with the forward Euler

(fully explicit) time discretization scheme. The initial fluxes are

$$\psi_{eqq}(0) = \frac{S_g}{4\pi N(0)\hat{\sigma}_g} \quad \longrightarrow \quad \phi_g(0) = \frac{S_g}{N(0)\hat{\sigma}_g}.$$

We now integrate the Bateman equation from $t = 0$ to $t = T$ using the time derivative evaluated at $t = 0$, and subsequently solve the transport equations for the $t = T$ fluxes:

$$\frac{dN}{dt} = -N(0)(\phi_1(0)\hat{\sigma}_1 + \phi_2(0)\hat{\sigma}_2) = -(S_1 + S_2) \quad \longrightarrow \quad \begin{cases} N(T) = N(0) - T \sum_g S_g \\ \psi_{eqq}(T) = \frac{S_g}{4\pi N(T)\hat{\sigma}_g} \end{cases}$$

We now put the problem in the context of the adjoint formalism given in Sec. 3.2. We first define a QOI: the total inventory (in moles) of the material at $t = T$. The expression for this QOI is

$$Q = \int_{\mathcal{D}} \frac{N(T)}{N_A} dV = \int_{\mathcal{D}} \int_0^\infty \int_{4\pi} \frac{N(T)}{N_A \int d\Omega \int dE} d\Omega dE dV = \frac{\mathbf{V} N(T)}{N_A}, \quad (\text{C.1})$$

where N_A is Avogadro's number in units of $\left[\frac{\text{atom}}{\text{mol}} \frac{\text{cm}^2}{\text{b}} \right]$. We see that for this discretization scheme, the QOI is not sensitive to the cross sections. The adjoint analysis should result in $\frac{dQ}{d\hat{\sigma}_g} = 0$.

From Eq. (C.1), we extract our definition for the functional R appearing in Eq. (3.47)

$$R \equiv \frac{N(T)}{N_A \int d\Omega \int dE},$$

which has units of $\left[\frac{\text{mol}}{\text{cm}^3 \text{MeV Ster}} \right]$.

The terminal condition for the adjoint densities is given by Eq. (3.51) (with $\hat{\psi}^\dagger = 0$

as

$$N^\dagger(T) = \frac{\partial}{\partial N} \langle g \rangle_{E,\mathcal{D},\Omega} = \frac{\mathbf{V}}{N_A},$$

which has units of $\left[\frac{\text{b}-\text{cm}-\text{mol}}{\text{atom}} \right]$. We can now solve for the terminal adjoint flux using the adjoint transport equation (Eq. (3.49)), which for this problem reduces to

$$H^\dagger \psi_{egq}^\dagger = N^\dagger \frac{\partial bN}{\partial \psi_{egq}}.$$

Operator b has the form of Eq. (2.25); to extract the expression for b of this problem, we write

$$\langle bN \rangle_{E,\mathcal{D},\Omega} = -N(\phi_1 \hat{\sigma}_1 + \phi_2 \hat{\sigma}_2) = \left\langle \frac{-\hat{\sigma}_g \psi_{egq} N}{\mathbf{V}} \right\rangle_{E,\mathcal{D},\Omega}$$

where we recall that ψ_{egq} must be converted to units of $\left[\frac{1}{\text{b}-\text{s}-\text{MeV}-\text{Ster}} \right]$. Then the operator is

$$bN = \frac{-\hat{\sigma}_g \psi_{egq} N}{\mathbf{V}} F_b^{\text{cm}},$$

which has units of $\left[\frac{\text{atom}}{\text{b}-\text{cm}^4-\text{s}-\text{MeV}-\text{Ster}} \right]$.

Our adjoint transport equation for adjoint flux at $t = T$ can now be written specifically as

$$\hat{\sigma}_g N(T) \psi_{egq}^\dagger(T) = \left(\frac{\mathbf{V}}{N_A} \right) \left(\frac{-\hat{\sigma}_g N(T) F_b^{\text{cm}}}{\mathbf{V}} \right),$$

yielding solution

$$\psi_{eqq}^\dagger(T) = -\frac{F_b^{cm}}{N_A}$$

which has units of $\left[\frac{\text{mol}}{\text{atom}}\right]$.

The next step is to integrate the adjoint Bateman equation, Eq. (3.48), to obtain the adjoint densities at $t = 0$. For adjoint consistency, this equation must be integrated with the forward Euler scheme, which for the reverse time integration, means that the derivatives are evaluated at $t = T$. For this problem, the adjoint Bateman equation reduces to

$$\frac{dN^\dagger}{dt} = \left\langle \psi^\dagger \frac{\partial H\psi}{\partial N} - b^\dagger N^\dagger \right\rangle_{E,\mathcal{D},\Omega} \quad (\text{C.2})$$

The first term inside the inner product on the right hand side, evaluated at $t = T$, is

$$\psi^\dagger(T) \frac{\partial H\psi(T)}{\partial N} = \left(-\frac{F_b^{cm}}{N_A} \right) \hat{\sigma}_g \psi_{eqq}(T) \quad (\text{C.3})$$

The second term inside the inner product requires a definition of b^\dagger , which we have shown is simply the transpose of the forward operator b . As we only have a single-term matrix,

$$b^\dagger N^\dagger = \left(-\frac{\hat{\sigma}_g \psi_{eqq}(T)}{\mathbf{V}} F_b^{cm} \right) \left(\frac{\mathbf{V}}{N_A} \right) = -\frac{\hat{\sigma}_g \psi_{eqq}(T) F_b^{cm}}{N_A}, \quad (\text{C.4})$$

which has units of $\left[\frac{\text{b-mol}}{\text{atom-s-MeV-Ster-cm}^2}\right]$. Subtracting Eq. (C.4) from Eq. (C.3), we see that the right hand side of Eq. (C.2) is zero, or

$$N^\dagger(0) = N^\dagger(T) = \frac{\mathbf{V}}{N_A}$$

Finally, the adjoint flux at $t = 0$ is also the same as at $t = T$, namely

$$\psi_{geq}^\dagger(0) = -\frac{1}{N_A}$$

The final task is to integrate the sensitivity equation, Eq. (3.52), using the forward and adjoint solutions derived above. This integration must be performed once for each $\hat{\sigma}_g$. The non-zero terms in the sensitivity equation are

$$\frac{dQ}{d\hat{\sigma}_g} = - \int_{t_0}^{t_f} \left\langle \psi^\dagger \left(\frac{\partial H\psi}{\partial \hat{\sigma}_g} \right) \right\rangle_{E,\mathcal{D},\Omega} dt + \int_{t_0}^{t_f} \left\langle N^\dagger \frac{\partial bN}{\partial \hat{\sigma}_g} \right\rangle_{E,\mathcal{D},\Omega} dt.$$

As described in Sec. 3.5, we compute this integral by appending a dummy system of equations to the adjoint system of equations. The forward Euler expression for this backwards integration is

$$\frac{dQ}{d\hat{\sigma}_g} = -T \left\{ -\left\langle \psi^\dagger(T) \left(\frac{\partial H\psi(T)}{\partial \hat{\sigma}_g} \right) \right\rangle_{E,\mathcal{D},\Omega} + \left\langle N^\dagger(T) \frac{\partial bN(T)}{\partial \hat{\sigma}_g} \right\rangle_{E,\mathcal{D},\Omega} \right\}$$

The first term on the right hand side is

$$-\left\langle \psi^\dagger(T) \frac{\partial H\psi(T)}{\partial \hat{\sigma}_g} \right\rangle_{E,\mathcal{D},\Omega} = -\left\langle \left(-\frac{F_b^{cm}}{N_A} \right) N(T) \psi_{eqq}(T) \right\rangle_{E,\mathcal{D},\Omega}, \quad (C.5)$$

which has units of $\left[\frac{mol}{b-s} \right]$. The second term is

$$\left\langle N^\dagger(T) \frac{\partial bN(T)}{\partial \hat{\sigma}_g} \right\rangle_{E,\mathcal{D},\Omega} = \left\langle \left(\frac{\mathbf{V}}{N_A} \right) \left(-\frac{F_b^{cm}}{\mathbf{V}} \right) \psi_{eqq}(T) N(T) \right\rangle_{E,\mathcal{D},\Omega} \quad (C.6)$$

which also has units of $\left[\frac{mol}{b-s} \right]$. We see that these terms cancel, giving the desired result of $\frac{dQ}{d\hat{\sigma}_g} = 0$.

This problem was run in PDT using the parameters and geometry given in Table

C.1.

Table C.1: Parameters for the “infinite medium”, pure absorber, 2 group adjoint verification problem

Parameter	Value
# Cells	1
Cell Volume, \mathbf{V}	$(2.0 \times 3.0 \times 2.0) \cdot 10^7 = 1.2\text{e+}22 \text{ cm}^3$
Spatial Discretization	PWLD (8 Elements)
Element Volume, V_e	1.5e+21 cm ³
Simulation Time, T	6.048e+05s = 1 week
Runge-Kutta Scheme	Explicit Euler
Initial Density, N_0	3 $\left[\frac{a}{bn - cm} \right]$
$\hat{\sigma}_1, \hat{\sigma}_2$	4.0 [bn], 1.0 [bn]
S_1, S_2	4.2e12 $\left[\frac{n}{\text{cm}^3 - \text{MeV} - s} \right]$, 4.2e12 $\left[\frac{n}{\text{cm}^3 - \text{MeV} - s} \right]$
Solver Tolerances	1.0e-07

Table C.2 gives a list of analytic expressions, their numerical value, the value computed with PDT, and the relative difference.

Table C.2: Key expressions, analytic values, and numerical values for the infinite medium, 2 group pure absorber adjoint depletion verification problem (ϵ = machine precision).

Description	Expression	Analytic Value	PDT Value	Relative Difference
Initial Group 1 Scalar Flux	$\frac{S_1}{N(0)\hat{\sigma}_1}$	3.50e11	3.4999999e+11	-6.4e-07%
Initial Group 2 Scalar Flux	$\frac{S_2}{N(0)\hat{\sigma}_2}$	1.40e12	1.3999999e+12	-2.6e-06%
Final Density	$N_0 - T \sum_g S_g$	2.9999949e+00	2.9999949e+00	ϵ
QOI	$\frac{\mathbf{V} N(T)}{N_A}$	5.9779299e+22	5.9779299e+22	ϵ
Final Group 1 Scalar Flux	$\frac{S_1}{N(T)\hat{\sigma}_1}$	3.5000059e+11	3.5000059e+11	-6.4e-07%
Final Group 2 Scalar Flux	$\frac{S_2}{N(T)\hat{\sigma}_2}$	1.4000024e+12	1.4000023e+12	-2.6e-06%
Terminal Adjoint Density, $N^\dagger(T)$	$\frac{\mathbf{V}}{N_A}$	1.9926467e+22	1.9926467e+22	ϵ
Gp. 1 Adjoint Trans. Src.	$-\frac{\hat{\sigma}_1 N(T) F_b^{cm}}{N_A}$	-1.9926433e-23	-1.9926433e-23	3.1e-06%
Gp. 2 Adjoint Trans. Src.	$-\frac{\hat{\sigma}_2 N(T) F_b^{cm}}{N_A}$	-4.9816083e-24	-4.9816084e-24	3.1e-06%
Gp. 1 Adjoint Scalar Flux, $t = T$	$-\frac{4\pi F_b^{cm}}{N_A}$	-2.0866947e-23	-2.0866948e-23	2.4e-06%
Gp. 2 Adjoint Scalar Flux, $t = T$	$-\frac{4\pi F_b^{cm}}{N_A}$	-2.0866947e-23	-2.0866948e-23	2.4e-06%

Table C.2 continued

Adj. Btmn Src Term 1: $\left\langle \text{Eq. (C.3)} \right\rangle_{E,\mathcal{D},\Omega}$	$\left\langle \psi^\dagger \frac{\partial H\psi}{\partial N} \right\rangle_{E,\mathcal{D},\Omega}$	-5.5794202e+10	-5.5794106e+10	-1.7e-04%
Adj. Btmn Src Term 2: $\left\langle \text{Eq. (C.4)} \right\rangle_{E,\mathcal{D},\Omega}$	$\left\langle b^\dagger N^\dagger \right\rangle_{E,\mathcal{D},\Omega}$	-5.5794202e-14	-5.5794106e+10	-1.7e-04%
First Term, Gp 1 Sens. Eq.: Eq. (C.5)	Eq. (C.5)	-2.0922790e+10	-2.0922790e+10	-1.3e-06%
Second Term, Gp 1 Sens. Eq.: Eq. (C.6)	Eq. (C.6)	-2.0922790e+10	-2.0922790e+10	-6.4e-07
First Term, Gp 2 Sens. Eq.: Eq. (C.5)	Eq. (C.5)	-8.3691161e+10	-8.3691157e+10	-5.1e-06%
Second Term, Gp 2 Sens. Eq.: Eq. (C.6)	Eq. (C.6)	-8.3691161e+10	-8.3691159e+10	-2.6e-06
Sensitivity w.r.t. $\hat{\sigma}_1$	$\frac{dQ}{d\hat{\sigma}_1}$	0.0	-8.1176259e+07	ϵ
Sensitivity w.r.t. $\hat{\sigma}_2$	$\frac{dQ}{d\hat{\sigma}_2}$	0.0	-1.2988200e+09	ϵ
Adj. Density, $N^\dagger(T)$	$\frac{\mathbf{V}}{N_A}$	1.9926467e+22	1.9926467e+22	ϵ
Gp. 1 Adjoint Scalar Flux, $t = 0$	$-\frac{4\pi F_b^{\text{cm}}}{N_A}$	-2.0866947e-23	-2.0866948e-23	2.4e-06%
Gp. 2 Adjoint Scalar Flux, $t = 0$	$-\frac{4\pi F_b^{\text{cm}}}{N_A}$	-2.0866947e-23	-2.0866948e-23	2.4e-06%

C.2 Infinite Medium, Source-Driven Problem with Scattering

We can also derive analytic solutions for a single-component, infinite-medium problem with down scattering and within-group scattering. We will use the analytic solutions to verify both sensitivity and error estimates. The governing equations are

$$\frac{dN}{dt} + F_b^{cm} N(\hat{\sigma}_{a,1}\phi_1 + \hat{\sigma}_{a,2}\phi_2) + \lambda N = 0$$

$$N\hat{\sigma}_{t,1}\psi_1 = \frac{S_1}{4\pi} + \frac{\hat{\sigma}_{s,11}\phi_1 N}{4\pi}$$

$$N\hat{\sigma}_{t,2}\psi_2 = \frac{S_2}{4\pi} + \frac{\hat{\sigma}_{s,22}\phi_2 N}{4\pi} + \frac{\hat{\sigma}_{s,12}\phi_1 N}{4\pi}$$

$$t \in [0, t_f]$$

$$N(0) = N_0$$

As we are in an infinite medium, $\phi_g = 4\pi\psi_g$, and the flux is the same for all angles and all spatial points. Also, the absorption cross sections are $\hat{\sigma}_{a,1} = \hat{\sigma}_{t,1} - \hat{\sigma}_{s,11} - \hat{\sigma}_{s,12}$ and $\hat{\sigma}_{a,2} = \hat{\sigma}_{t,2} - \hat{\sigma}_{s,22}$. We find the following flux solutions:

$$\begin{aligned} \psi_1 &= \frac{S_1}{4\pi N \hat{\sigma}_{R,1}} & \rightarrow \quad \phi_1 &= \frac{S_1}{N \hat{\sigma}_{R,1}} \\ \psi_2 &= \frac{S_2}{4\pi N \hat{\sigma}_{R,2}} + \frac{S_1 \hat{\sigma}_{s,12}}{4\pi \hat{\sigma}_{R,1} \hat{\sigma}_{R,2} N} & \rightarrow \quad \phi_2 &= \frac{S_2}{N \hat{\sigma}_{R,2}} + \frac{S_1 \hat{\sigma}_{s,12}}{\hat{\sigma}_{R,1} \hat{\sigma}_{R,2} N}, \end{aligned}$$

where $\hat{\sigma}_{R,1} = \hat{\sigma}_{1,t} - \hat{\sigma}_{s,11}$ and $\hat{\sigma}_{R,2} = \hat{\sigma}_{t,2} - \hat{\sigma}_{s,22} = \hat{\sigma}_{a,2}$. Substituting the flux solutions into the Bateman equation, we find

$$\begin{aligned}
0 &= \frac{dN}{dt} + F_b^{cm} N \left(\hat{\sigma}_{a,1} \frac{S_1}{N \hat{\sigma}_{R,1}} + \hat{\sigma}_{a,2} \left[\frac{S_2}{N \hat{\sigma}_{R,2}} + \frac{S_1 \hat{\sigma}_{s,12}}{N \hat{\sigma}_{R,1} \hat{\sigma}_{R,2}} \right] \right) + \lambda N \\
&= \frac{dN}{dt} + F_b^{cm} \left(\frac{S_1 \hat{\sigma}_{a,1}}{\hat{\sigma}_{R,1}} + \frac{S_1 \hat{\sigma}_{s,12} \hat{\sigma}_{a,2}}{\hat{\sigma}_{R,1} \hat{\sigma}_{R,2}} + \frac{S_2 \hat{\sigma}_{a,2}}{\hat{\sigma}_{R,2}} \right) + \lambda N \\
&= \frac{dN}{dt} + F_b^{cm} \left(\frac{S_1 \hat{\sigma}_{a,1}}{\hat{\sigma}_{R,1}} + \frac{S_1 \hat{\sigma}_{s,12}}{\hat{\sigma}_{R,1}} + S_2 \right) \\
&= \frac{dN}{dt} + F_b^{cm} (S_1 + S_2) + \lambda N \\
&\equiv \frac{dN}{dt} + \alpha + \lambda N
\end{aligned}$$

The analytic solution to the Bateman equation is

$$N(t) = \left[\frac{\alpha}{\lambda} + N_0 \right] \exp(-\lambda t) - \frac{\alpha}{\lambda}.$$

Define our QOI as the total number of mols of the component at $t = t_f$,

$$Q = \left\langle \frac{1}{\langle \rangle_{E,\Omega}} \frac{N}{N_A} \right\rangle_{E,\mathcal{D},\Omega} = \frac{N(t_f)V}{N_A}.$$

Thus we have an analytic solution for our QOI

$$Q = \frac{V}{N_A} \left[\frac{\alpha}{\lambda} + N_0 \right] \exp(-\lambda t_f) - \frac{\alpha V}{\lambda N_A} \quad (\text{C.7})$$

Note that this QOI is not sensitive to the cross sections: $\frac{dQ}{d\hat{\sigma}_*} = 0$. We have an analytic sensitivity w.r.t the decay constant, λ :

$$\frac{dQ}{d\lambda} = -\frac{V}{N_A} \left[\frac{t_f \alpha}{\lambda} + t_f N_0 + \frac{\alpha}{\lambda^2} \right] \exp(-\lambda t_f) + \frac{\alpha V}{N_A \lambda^2} \quad (\text{C.8})$$

Consider solving our Bateman equation (which is not dependent on the flux) using a series of explicit Euler steps. Let the time step be fixed at T and denote the solution $N(t = jT)$ as N_j . The procession is

$$\begin{aligned}
N_1 &= N_0 - T(\alpha + \lambda N_0) \\
&= (1 - \lambda T)N_0 - \alpha T \\
N_2 &= N_1 - T(\alpha + \lambda N_1) \\
&= (1 - \lambda T)N_1 - \alpha T \\
&= (1 - \lambda T)^2 N_0 - \alpha T(1 - \lambda T) - \alpha T \\
N_3 &= N_2 - T(\alpha + \lambda N_2) \\
&= (1 - \lambda T)N_2 - \alpha T \\
&= (1 - \lambda T)^3 N_0 - \alpha T(1 - \lambda T)^2 - \alpha T(1 - \lambda T) - \alpha T \\
&\dots \\
N_j &= (1 - \lambda T)^j N_0 - \alpha T \sum_{i=0}^{j-1} (1 - \lambda T)^i.
\end{aligned}$$

The expressions for our discrete QOI, $\tilde{Q}_j = \tilde{Q}(t = jT)$, and its sensitivity to λ are

$$\tilde{Q}_j = \frac{V}{N_A} \left[(1 - \lambda T)^j N_0 - \alpha T \sum_{i=0}^{j-1} (1 - \lambda T)^i \right] \quad (\text{C.9})$$

$$\frac{d\tilde{Q}_j}{d\lambda} = \frac{V}{N_A} \left[-jTN_0(1 - \lambda T)^{j-1} + \alpha T^2 \sum_{i=0}^{j-2} (i+1)(1 - \lambda T)^i \right] \quad (\text{C.10})$$

First, we verify that this time marching scheme is giving us the expected result,

$\tilde{Q} = Q + \mathcal{O}(T)$. If we let $t_f = jT$ and expand Eq. (C.7) about small T , we find

$$Q = \frac{V}{N_A} \left[N_0 - (\alpha + N_0\lambda)T + \frac{1}{2}(\alpha\lambda + \lambda^2 N_0)T^2 - \frac{1}{6}(N_0\lambda^3 + \alpha\lambda^2)T^3 \right] + \mathcal{O}(T^4) \quad (\text{C.11})$$

Next we expand Eq. (C.9) to $\mathcal{O}(T^3)$:

$$\begin{aligned} \tilde{Q} &= \frac{V}{N_A} \left[N_0 - j\lambda N_0 T + \sum_{i=0}^{j-1} i\lambda N_0 T^2 - \alpha T \sum_{i=0}^{j-1} (1 - i\lambda T) \right] + \mathcal{O}(T^3) \\ &= \frac{V}{N_A} \left[N_0 - j\lambda N_0 T + \sum_{i=0}^{j-1} i\lambda N_0 T^2 - j\alpha T + \sum_{i=0}^{j-1} i\alpha\lambda T^2 \right] + \mathcal{O}(T^3) \end{aligned} \quad (\text{C.12})$$

By inspection of Eqs. (C.11) and (C.12), we indeed see that the expansions of Q and \tilde{Q} agree through $\mathcal{O}(T)$ terms. Similarly, we expand our expression for the analytic sensitivity

$$\begin{aligned} \frac{dQ}{d\lambda} &= -\frac{V}{N_A} \left[\frac{jT\alpha}{\lambda} + jTN_0 + \frac{\alpha}{\lambda^2} \right] \left[1 - j\lambda T + \frac{1}{2}j^2\lambda^2 T^2 \right] + \frac{\alpha V}{N_A \lambda^2} + \mathcal{O}(T^3) \\ &= -\frac{V}{N_A} \left[\frac{j\alpha T}{\lambda} + jTN_0 + \frac{\alpha}{\lambda^2} - j^2\alpha T^2 - j^2\lambda N_0 T^2 \right. \\ &\quad \left. - \frac{j\alpha T}{\lambda} + \frac{1}{2}j^2\alpha T^2 \right] + \frac{\alpha V}{N_A \lambda^2} + \mathcal{O}(T^3) \\ &= -\frac{V}{N_A} \left[jN_0 T - \left[\frac{1}{2}j^2\alpha + j^2\lambda N_0 \right] T^2 \right] + \mathcal{O}(T^3) \end{aligned} \quad (\text{C.13})$$

and the discrete sensitivity

$$\frac{d\tilde{Q}}{d\lambda} = \frac{V}{N_A} \left[-jN_0 T + j(j-1)N_0\lambda T^2 + \alpha T^2 \sum_{i=0}^{j-2} (i+1) \right] \quad (\text{C.14})$$

Again, by inspection of Eqs. (C.13) and (C.14) we see that the discrete sensitivity matches the analytic sensitivity through $\mathcal{O}(T)$ terms.

This problem was run in PDT using the parameters and geometry given in Table C.3.

Table C.3: Parameters for the “infinite medium”, 2 group adjoint verification problem with within-group and down scattering

Parameter	Value
# Cells	1
Cell Volume, \mathbf{V}	$(2.0 \times 3.0 \times 2.0) \cdot 10^7 = 1.2\text{e}+22 \text{ cm}^3$
Spatial Discretization	PWLD (8 Elements)
Element Volume, V_e	$1.5\text{e}+21 \text{ cm}^3$
Simulation Time, t_f	5 Cycles of $5.0\text{e}9 \text{ s}$
Time Steps, T	Varying
Initial Density, N_0	$3.0 \left[\frac{a}{bn-cm} \right]$
$\hat{\sigma}_{t,1}, \hat{\sigma}_{t,2}$	4.0, 2.2 [bn]
$\hat{\sigma}_{s,11}, \hat{\sigma}_{s,12}, \hat{\sigma}_{s,21}, \hat{\sigma}_{s,22}$	0.3, 1.2, 0.0, 1.2 [bn]
S_1, S_2	$4.2\text{e}14 \left[\frac{n}{\text{cm}^3-\text{MeV-s}} \right], 4.2\text{e}14 \left[\frac{n}{\text{cm}^3-\text{MeV-s}} \right]$
Solver Tolerances	1.0e-12

First we show that PDT is indeed computing the predicted explicit Euler result. Table C.4 gives ϵ , defined as

$$\epsilon = \frac{\tilde{Q}_{\text{PDT}} - \tilde{Q}_{\text{Pred}}}{\tilde{Q}_{\text{Pred}}}$$

for each of the different time steps that were tested. Here, \tilde{Q}_{PDT} is the PDT result and \tilde{Q}_{Pred} is the evaluation of Eq. (C.9). We see that the relative difference is very small.

Table C.4: Comparison of the predicted Explicit Euler result, Eq. (C.9), and the result computed in PDT

Time Step (s)	Difference, ϵ
1.5625e+08	8.9598e-09
3.125e+08	8.9225e-09
6.25e+08	8.8491e-09
1.25e+09	8.7064e-09
2.50e+09	8.4307e-09
5.00e+09	7.9215e-09

Next we verify that our QOI prediction and its sensitivity to λ are achieving the order of accuracy predicted by theory. Above, we showed that the explicit Euler method should converge linearly with the time step T . We also test the modified Euler Runge-Kutta scheme (see Eq. (E.2) in Appendix E), which should achieve second order accuracy in these predictions. Figure C.1 shows the error in predicting the QOI as a function of the time step. Figure C.2 shows the error in predicting the sensitivity of the QOI with respect to the decay constant λ .

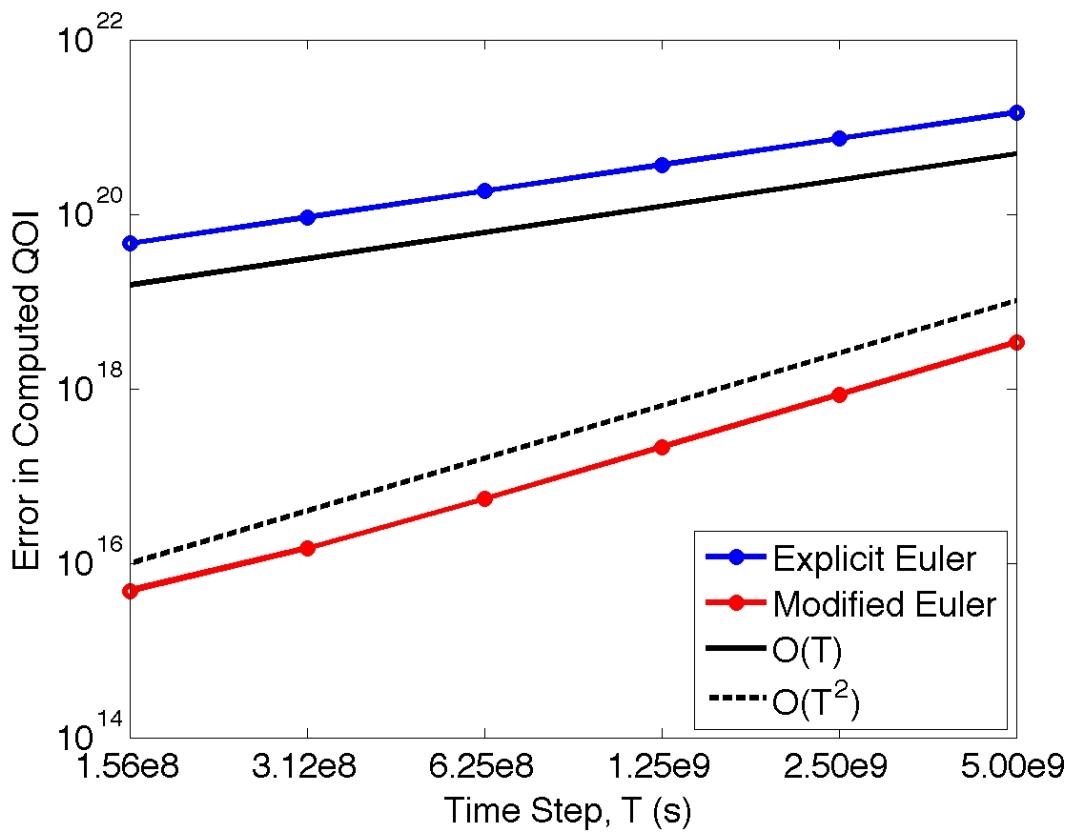


Figure C.1: The figure gives the error in predicting the inventory QOI as a function of time-step size. The explicit Euler method converges as $\mathcal{O}(T)$ and the modified Euler method converges as $\mathcal{O}(T^2)$, as predicted by theory.

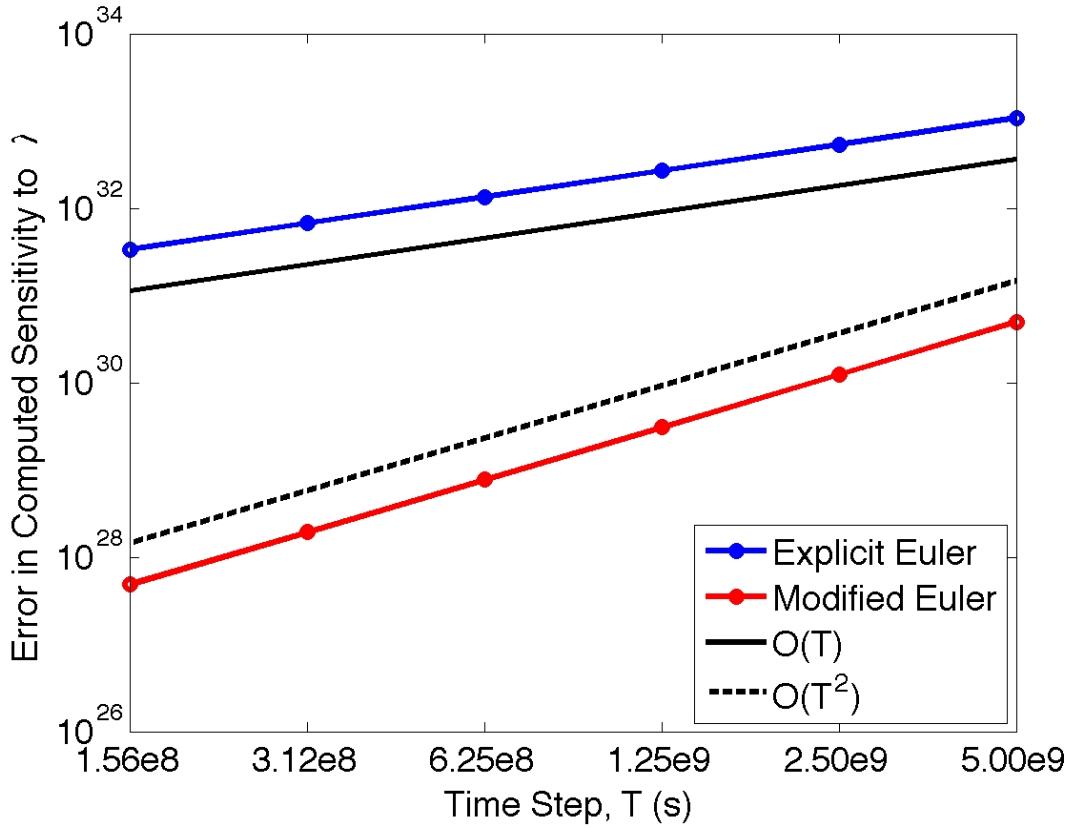


Figure C.2: The figure gives the error in predicting the sensitivity of the QOI with respect to λ as a function of time-step size. The explicit Euler method converges as $\mathcal{O}(T)$ and the modified Euler method converges as $\mathcal{O}(T^2)$, as predicted by theory.

This verification problem is also useful for testing the prediction of the global discretization error. First, consider solving for a single time-step, $t_f = T$. By Eq. (C.7), the analytic value of the QOI is

$$Q(T) = \frac{V}{N_A} \left[\frac{\alpha}{\lambda} + N_0 \right] \exp(-\lambda T) - \frac{V\alpha}{N_A \lambda}.$$

The explicit Euler solution over the time step is simply

$$N^{(1)}(T) = N_0 - \alpha T - \lambda N_0 T, \quad (\text{C.15})$$

where the superscript (p) indicates an order- (p) solution. Using this or Eq. (C.9) with $j = 1$, we have for our discrete QOI

$$\tilde{Q}(T) = \frac{V}{N_A} [N_0 - \alpha T - \lambda N_0 T].$$

We will now follow our formalism given in section 3.1.2 for computing an estimate for $\Delta Q = Q - \tilde{Q}$, defined in this case as the error in predicting the QOI due to truncation in the time discretization. For this, we will consider only $r_1^d(t)$, which is a residual error made in evaluating the differential governing equations. In this case, Eq. (3.53) simplifies to

$$\Delta Q = - \int_{t_0}^{t_f} N^\dagger(s) r_1^d(s) ds. \quad (\text{C.16})$$

The residual r_1^d is defined by subtracting the exact Bateman equation from the approximate Bateman equation as

$$\begin{aligned} r_1^d(T) &= B(\phi(T))N(T) - B(\tilde{\phi}(T))\tilde{N}(T) - \left(\frac{dN}{dt} - \frac{d\tilde{N}}{dt} \right) \\ &= (-\alpha - \lambda N(T)) - (-\alpha - \lambda \tilde{N}(T)) - \left(\frac{N(T) - N_0}{T} - \frac{\tilde{N}(T) - N_0}{T} \right) \\ &= -\lambda [N(T) - \tilde{N}(T)] - \frac{N(T) - \tilde{N}(T)}{T}. \end{aligned} \quad (\text{C.17})$$

In this expression, N is the true (or continuous) density vector, and \tilde{N} is the approximate (discrete) density vector. We do not have the continuous solution; we can, however, use our Heun-Euler (see Eq. (E.4) in Appendix E) embedded Runge-Kutta

method to obtain a second-order accurate estimate for N . This method gives

$$N^{(2)}(T) = N_0 - \frac{T}{2} (\alpha + \lambda \hat{N}_1) - \frac{T}{2} (\alpha + \lambda \hat{N}_2)$$

where

$$\hat{N}_1 = N_0$$

$$\hat{N}_2 = N_0 - \alpha T - \lambda N_0 T.$$

After some manipulation, we find our second-order accurate solution for N is

$$N^{(2)}(T) = N_0 - \alpha T - \lambda N_0 T + \frac{1}{2} \alpha \lambda T^2 + \frac{1}{2} \lambda^2 N_0 T^2 \quad (\text{C.18})$$

If we take $N^{(2)}$ as our estimate for N and substitute Eqs. (C.15) and (C.18) into Eq. (C.17), we find the following expression for our residual:

$$r_1^d(T) = -\frac{1}{2} \lambda T (1 + \lambda T) (\alpha + \lambda N_0).$$

By Eq. (3.51), the terminal value of the adjoint density is $N^\dagger(T) = \frac{V}{N_A}$.

At this point, we must decide how to integrate Eq. (C.16). If we choose an explicit, first order rule (that is, a rule consistent with the integration of the adjoint equations), we find

$$\Delta Q_{\text{pred}} = \frac{V}{2N_A} \lambda T^2 (1 + \lambda T) (\alpha + \lambda N_0).$$

We can check the accuracy of this estimate by expanding ΔQ_{pred} and $\Delta Q_{\text{analytic}} \equiv$

$Q - \tilde{Q}$ in terms of T :

$$\begin{aligned}\Delta Q_{\text{pred}} &= \frac{V}{2N_A} (\lambda T^2 + \lambda^2 T^2) (\alpha + \lambda N_0) \\ &= \frac{V}{2N_A} (\alpha \lambda T^2 + \lambda^2 N_0 T^2 + \alpha \lambda^2 T^3 + \lambda^3 N_0 T^3) \\ \Delta Q_{\text{analytic}} &= \frac{V}{N_A} \left[\frac{\alpha}{\lambda} + N_0 \right] \left[1 - \lambda T + \frac{1}{2} \lambda^2 T^2 + \mathcal{O}(T^3) \right] - \frac{V \alpha}{\lambda N_A} - \frac{V}{N_A} [N_0 - \alpha T - \lambda N_0 T] \\ &= \frac{V}{2N_A} [\alpha \lambda T^2 + \lambda^2 N_0 T^2] + \mathcal{O}(T^3).\end{aligned}$$

Note that the prediction of the error and the actual error match through $\mathcal{O}(T^2)$.

We now move back to our problem of multiple time steps and multiple cycles. We expect the accuracy of the error prediction to drop by a factor of T when integrating over multiple time steps. Figure C.3 shows that indeed the error in the error prediction is decreasing as $\mathcal{O}(T^2)$ for this test problem.

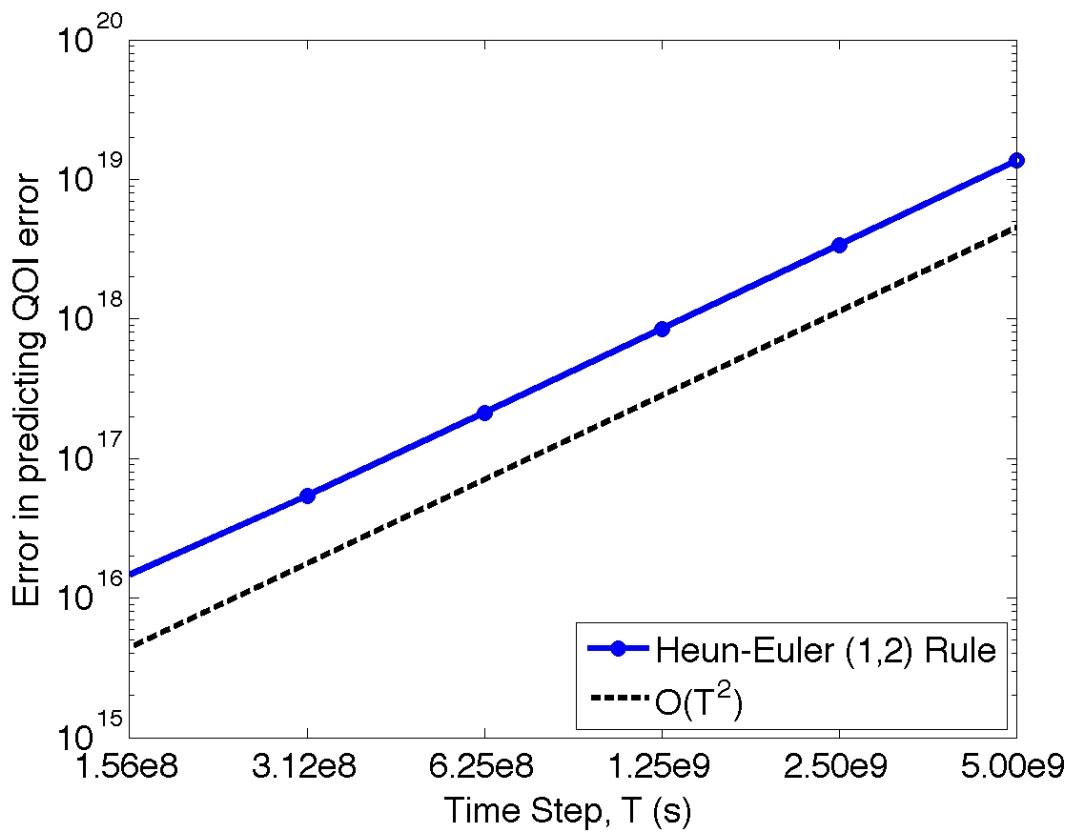


Figure C.3: The figure gives the error in predicting the error in the QOI as a function of time-step size. For this problem, the error is converging as $\mathcal{O}(T^2)$.

APPENDIX D

THE FUNDAMENTAL SOLUTION MATRIX METHOD APPLIED TO FORWARD AND ADJOINT SYSTEMS

In this appendix we review the fundamental solution matrix and its application in solving a linear system of equations and the corresponding adjoint system[61]. Consider the general, homogeneous system of ordinary differential equations

$$\frac{dx}{dt} = A(t)x, \quad t \in [t_0, t_f], \quad x \in \mathbb{R}^n \quad (\text{D.1})$$

Under a common set of assumptions, this system has an infinitude of solutions, each corresponding to a unique initial condition $x(t_0) = x_0$. Define the notation $x(t; t_0, x_i)$ to be the solution $x(t)$ corresponding to the initial condition $x(t_0) = x_i$. This is, in effect, a mapping from $x_i \in \mathbb{R}^n$ to $x(t) \in \mathbb{R}^n$, which we will write as $x(t) = \mathcal{C}(t; t_0)x_i$. If x_1 and x_2 are two linearly independent initial conditions, then it follows that

$$\mathcal{C}(t; t_0)(\alpha_1 x_1 + \alpha_2 x_2) = x(t; t_0, \alpha_1 x_1 + \alpha_2 x_2)$$

and

$$\alpha_1 \mathcal{C}(t; t_0)x_1 + \alpha_2 \mathcal{C}(t; t_0)x_2 = \alpha_1 x(t; t_0, x_1) + \alpha_2 x(t; t_0, x_2)$$

are equivalent. Thus, the mapping $\mathcal{C}(t; t_0)$ is linear. In particular, note that

$$x(t_0) = \mathcal{C}(t_0; t_0)x_0 \quad \rightarrow \quad \mathcal{C}(t_0; t_0) = I_n.$$

A final essential property is that for all t, s, u , we have

$$\mathcal{C}(t; s)\mathcal{C}(s; u) = \mathcal{C}(t; u)$$

This can be shown by writing

$$\mathcal{C}(t; s)\mathcal{C}(s; u)x_0 = x(t; s, \mathcal{C}(s; u)x_0) = x(t; s, x(s; u, x_0)) = x(t; u, x_0) = \mathcal{C}(t; u)x_0$$

Then, for $s = t_0$ and $u = t$, we can write

$$\mathcal{C}(t; t_0)\mathcal{C}(t_0, t) = \mathcal{C}(t; t) = I_n,$$

revealing that $\mathcal{C}(t; t_0)$ is invertible, and its inverse is $\mathcal{C}(t_0; t)$.

Now we are ready to proceed with forming solutions to Eq. (D.1) and its adjoint. First, define the matrix $\Phi \in \mathbb{R}^{n \times n}$ as the matrix whose columns are a set of n linearly independent solutions to Eq. (D.1), $\Phi \equiv [x_1(t), \dots, x_n(t)]$. This matrix is called a fundamental solution matrix, and it satisfies

$$\frac{d\Phi}{dt} = A(t)\Phi(t)$$

Theory of superposition gives that for any vector $v \in \mathbb{R}^n$, the vector

$$y = \Phi v$$

is a solution to Eq. (D.1). In particular, consider solving for a set of coefficients to

satisfy the initial condition

$$x_0 = \Phi(0)v \quad \rightarrow \quad v = \Phi^{-1}(0)x_0,$$

and the solution to Eq. (D.1) with initial condition $x(t_0) = x_0$ is

$$x(t) = \Phi(t)\Phi^{-1}(0)x_0.$$

Note that this is immediately related to our mapping from before: $\mathcal{C}(t, t_0) = \Phi(t)\Phi^{-1}(0)$.

Now consider the adjoint of Eq. (D.1),

$$\frac{dx^\dagger}{dt} = -A^\dagger(t)x^\dagger, \tag{D.2}$$

where A^\dagger is such that $\langle x^\dagger, Ax \rangle = \langle x, A^\dagger x^\dagger \rangle$. Let $\mathcal{C}^\dagger(t; t_0)$ be its corresponding linear transformation. To relate $\mathcal{C}^\dagger(t; t_0)$ and $\mathcal{C}(t; t_0)$, we first show that the inner product $\langle x, x^\dagger \rangle$ is a constant:

$$\begin{aligned} \frac{d}{dt} \langle x, x^\dagger \rangle &= \left\langle \frac{dx}{dt}, x^\dagger \right\rangle + \left\langle x, \frac{dx^\dagger}{dt} \right\rangle \\ &= \langle Ax, x^\dagger \rangle - \langle x, A^\dagger x^\dagger \rangle = 0. \end{aligned}$$

Now note that $[\mathcal{C}^\dagger(t; t_0)]^T$ is a matrix whose rows are solutions to the adjoint system.

Thus,

$$\frac{d}{dt} \left\langle [\mathcal{C}^\dagger(t; t_0)]^T \mathcal{C}(t, t_0) \right\rangle = 0,$$

because this matrix multiplication is simply a series of dot products of adjoint and forward solution vectors. Now, using a previous result, namely $[\mathcal{C}^\dagger(t_0; t_0)]^T \mathcal{C}(t_0, t_0) =$

$I_n I_n = I_n$, and the preceding equation, we can write

$$\left\langle \left[\mathcal{C}^\dagger(t; t_0) \right]^T \mathcal{C}(t, t_0) \right\rangle = \left\langle I_n \right\rangle \quad \forall t$$

This is satisfied if the following is true:

$$\left[\mathcal{C}^\dagger(t; t_0) \right]^T = \mathcal{C}^{-1}(t; t_0)$$

or

$$\mathcal{C}^\dagger(t; t_0) = \mathcal{C}^{-T}(t; t_0).$$

If this is true,

$$\mathcal{C}^\dagger(t; t_0) = \mathcal{C}^{-T}(t; t_0) = \left[\Phi(t) \Phi^{-1}(t_0) \right]^{-T} = \Phi^{-T}(t) \Phi^T(0)$$

Adjoint equations typically have terminal conditions. Thus, the solution to Eq. (D.2) with terminal condition $x^\dagger(t_f) = x_f^\dagger$ is

$$x^\dagger(s) = \Phi^{-T}(s) \Phi^T(t_f) x_f^\dagger.$$

Finally, to support some theory in this dissertation, we will need to use the preceding formalism to solve equations of the form

$$\frac{dx}{dt} = A(t)x + f(t)$$

Following the theory in [61], we make a change of variables $x(t) = \mathcal{C}(t; t_0)y(t)$, note

that $y(t_0) = x(t_0)$, and write

$$\begin{aligned}\frac{dx}{dt} &= \frac{d\mathcal{C}(t; t_0)}{dt}y(t) + \mathcal{C}(t; t_0)\frac{dy}{dt} \\ &= A(t)\mathcal{C}(t; t_0)y(t) + f(t).\end{aligned}$$

From this we extract $\mathcal{C}(t; t_0)\frac{dy}{dt} = f(t)$, or $\frac{dy}{dt} = \mathcal{C}^{-1}(t; t_0)f(t) = \mathcal{C}(t_0; t)f(t)$. If we integrate this equation from y_0 :

$$y(t) = y_0 + \int_{t_0}^t \mathcal{C}(t_0; s)f(s)ds$$

and transform back to the variable x :

$$x(t; t_0, x_0) = \mathcal{C}(t; t_0)x_0 + \mathcal{C}(t; t_0) \int_{t_0}^t \mathcal{C}(t_0; s)f(s)ds$$

and find that the solution to the inhomogeneous equations can be related to Φ as follows

$$x(t; t_0, x_0) = \Phi(t)\Phi^{-1}(0)x_0 + \int_{t_0}^t \Phi(t)\Phi^{-1}(s)f(s)ds \quad (\text{D.3})$$

APPENDIX E

LISTING OF RUNGE-KUTTA METHODS

The following is a list of common Runge-Kutta methods, some of which are used in the analysis and numerical examples of this dissertation.

- Explicit Euler

- Number of Stages: 1
- Order of accuracy: 1
- Explicit method
- Embedded Method? No
- Butcher tableau:

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array} \quad (\text{E.1})$$

- Modified Euler

- Number of Stages: 2
- Order of accuracy: 2
- Explicit method
- Embedded Method? No

- Butcher tableau:

$$\begin{array}{c|cc} 0 & 0 \\ \hline 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \quad (\text{E.2})$$

- Classic Fourth Order (RK4)

- Number of Stages: 4
- Order of accuracy: 4
- Explicit method
- Embedded Method? No
- Butcher tableau:

$$\begin{array}{c|cccc} 0 & 0 & & & \\ \hline \frac{1}{2} & \frac{1}{2} & 0 & & \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \\ \hline 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \quad (\text{E.3})$$

- Heun-Euler

- Number of Stages: 1/2
- Order of Accuracy: 1/2
- Explicit Method
- Embedded Method? Yes

- Butcher tableau:

$$\begin{array}{c|cc}
 0 & 0 & 0 \\
 1 & 1 & 0 \\
 \hline
 & 1 & 0 \\
 & \frac{1}{2} & \frac{1}{2}
 \end{array} \tag{E.4}$$

- Implicit Euler

- Number of Stages: 1
- Order of accuracy: 1
- Implicit method
- Embedded Method? No
- Butcher tableau:

$$\begin{array}{c|c}
 1 & 1 \\
 \hline
 & 1
 \end{array} \tag{E.5}$$

- Implicit “Midpoint” Rule

- Number of Stages: 1
- Order of accuracy: 1
- Implicit method
- Embedded Method? No
- Butcher tableau:

$$\begin{array}{c|c}
 \frac{1}{2} & \frac{1}{2} \\
 \hline
 & 1
 \end{array} \tag{E.6}$$

APPENDIX F

LISTING OF DEPLETION CHAINS

F.1 Synthetic Two Group, Four Component Model

Group boundaries (eV): 5.0, 4.0, 0.0

Depletion mechanisms:

- U-235 (n,removal) (nothing)
- U-235 (n,fission) Fe-56 (83%)
- U-238 (n,removal) (nothing)
- U-238 (n, γ) U-239
- U-238 (n,fission) Fe-56 (13%)
- U-239 (n,removal) (nothing)
- U-239 (n,fission) Fe-56 (50%)
- Fe-56 (n,removal) (nothing)

Table F.1: List of parameters and values for the synthetic, 2-group, 4-component depletion model

Parameter	U-235	U-238	U-239	Fe-56
$\sigma_{t,1}$ [b]	1.8	1.8	2.0	2.6
$\sigma_{t,2}$ [b]	8.3	12.3	9.3	1.3
$\sigma_{f,1}$ [b]	0.7	0.15	0.0	0.0
$\sigma_{f,2}$ [b]	5.0	6.1	6.0	0.0
$\sigma_{\gamma,1}$ [b]	0.4	0.8	0.8	0.0
$\sigma_{\gamma,2}$ [b]	1.3	3.3	0.3	0.0
E_f [MeV]	190.3	210.0	220.3	0.0
ν_1	2.2	2.2	2.2	0.0
ν_2	2.2	2.2	2.2	0.0
χ_1	0.9	0.8	0.95	n/a
χ_2	0.1	0.2	0.05	n/a
$\sigma_{s,1 \rightarrow 1}$ [b]	0.05	0.15	0.45	0.50
$\sigma_{s,1 \rightarrow 2}$ [b]	0.2	0.2	0.2	0.04
$\sigma_{s,2 \rightarrow 1}$ [b]	0.0	0.0	0.2	0.8
$\sigma_{s,2 \rightarrow 2}$ [b]	2.0	1.0	1.4	0.2

F.2 33 group, fast spectrum cross sections from Argonne National Laboratory

Michael Smith, Oleg Roderick, and Mihai Anitescu, staff members at Argonne National Laboratory (ANL), generated this set of cross sections for use in a fast-spectrum, traveling wave reactor simulation. They are spatially self-shielded, 33-group cross sections with data for total, capture, n-2n, n-proton, n-deuteron, n-triton, n-alpha, fission, neutrons per fission, fission spectrum, and scattering with three moments. The nuclides provided are listed in Fig. F.1, and the nuclides used in this dissertation are enclosed by the red boxes.

*	*	U234A	U235A	U236A	U238A	N237A	N238A	N239A	P236A	P237A
P238A	P239A	P240A	P241A	P242A	A241A	A24MA	A242A	A243A	C242A	
C243A	C244A	C245A	C246A	FE56A	NA23A	BR81A	KR83A	KR84A	RB85A	
RB87A	Y-89A	SR90A	ZR90A	ZR91A	ZR92A	ZR93A	ZR94A	ZR96A	M095A	
M096A	M097A	M098A	TC99A	M0100A	RU100A	RU101A	RU102A	RH103A	RU104A	
PD104A	PD105A	RU106A	PD106A	PD107A	PD108A	AG109A	PD110A	CD110A	CD111A	
CD112A	CD113A	CD114A	IN115A	SN117A	SN118A	SB121A	SB123A	SB125A	TE125A	
I-127A	TE128A	I-129A	TE130A	XE130A	XE131A	XE132A	CS133A	XE134A	CS134A	
BA134A	XE135A	CS135A	CS137A	BA137A	BA138A	LA139A	CE140A	PR141A	CE142A	
ND143A	ND144A	ND145A	ND146A	PM147A	SM147A	ND148A	SM148A	SM149A	ND150A	
SM150A	SM151A	EU151A	SM152A	EU152A	EU153A	SM154A	EU154A	GD154A	EU155A	
GD155A	GD156A	GD157A	GD158A	TB159A	DUMMY1	DUMMY2	GE73_5	GE74_5	GE76_5	
AS75_5	SE76_5	SE77_5	SE78_5	SE80_5	SE82_5	BR79_5	KR80_5	KR82_5	KR86_5	
RB86_5	SR86_5	SR87_5	SR88_5	NB93_5	NB94_5	M094_5	RU99_5	AG1077	CD1087	
CD1167	IN1137	SN1157	SN1167	SN1197	SN1207	SN1227	SN1247	SN1267	SB1267	
TE1227	TE1237	TE1247	TE1267	I130_7	I131_7	I135_7	XE1287	XE1367	CS1367	
BA1357	BA1367	PR1427	ND1427	GD1527	GD1607	TB1607	DY1607	DY1617	DY1627	
DY1637	DY1647	H01657	ER1667	ER1677	LTH32	LU-33	LU-34	LU-35	LU-36	
LU-38	LNP37	LPU38	LPU39	LPU40	LPU41	LPU42	LAM41	LAM42	LAM43	
LCM42	LCM43	LCM44	LCM45	LCM46	MTH32	MU-33	MU-34	MU-35	MU-36	
MU-38	MNP37	MPU38	MPU39	MPU40	MPU41	MPU42	MAM41	MAM42	MAM43	
MCM42	MCM43	MCM44	MCM45	MCM46						

Figure F.1: List of nuclides in the 33g ANL fast-spectrum cross section dataset

The cross sections were generated by the MC² Monte Carlo transport code using fast-spectrum averaging, ENDF-VII data, and a reference composition from benchmark solutions to the traveling wave reactor problem. As the figure indicates, the cross sections that contributed to this work are the major actinides and the lumped fission products. There is one lumped fission product for each actinide that could fission (e.g. MU-35 corresponds to Uranium-235), and that actinide produces one such fission product per fission event.