Robust and Efficient Methods in Transient Whole-Core Neutron Transport Calculations

by

Qicang Shen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences and Scientific Computing)
in the University of Michigan
2021

Doctoral Committee:

Professor Thomas Downar, Co-Chair
Professor Brendan Kochunas, Co-Chair
Assistant Research Scientist Yuxuan Liu
Professor Shravan Veerapaneni
Professor Yunlin Xu, Purdue University
Professor Won Sik Yang

qicangsh@umich.edu

ORCID ID: 0000-0002-3904-8680

ACKNOWLEDGMENTS

First, I would like to thank my advisors, Brendan Kochunas and Thomas Downar for their help, patience and guidance, so that the work in this dissertation has some real applications and can be delivered to the public. Professor Downar, I really appreciate your passion and invitation that made me decide to come to UM. This is a decision that I will never regret. Brendan, I feel so grateful for your bearing with my poor English, your trust, your suggestions and your novel ideas. Working with you is a really wonderful experience.

Second, I would like to thank my committee for their help. I would like to thank Professor Yunlin Xu for being my mentor to guide me in the field of nuclear reactor physics when I first came to UM; Professor Yang for your rigorousness in teaching and research that really impressed me; Professor Veerapaneni for your mathematical suggestions and for all that I have learned in your courses; Dr. Yuxuan Liu for discussions that led me to think more about my research.

I must also acknowledge the Consortium for Advanced Simulation of Light Water Reactors (CASL) (www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract No. DE-AC05-00OR22725 for supporting my research. This research made use of the resources of the High Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517.

Next, I am also incredibly indebted to my colleagues of the Nuclear Reactor Analysis & Methods (NURAM) group on the Fourth Floor in ERB including Dr. Ben Yee, Dr. Mike Jarrett, Dr. Andrew Fitzgerald, Dr. Volkan Selker, Dr. Andrew Ward, Dr. Sooyoung Choi, Dan Jabby and Zack Dodson. Thanks for your suggestions and help on my research, and your humor that lets the Fourth floor full of joyfulness.

I am also thankful to my friends that I met here. It is my fortune to have you in my life. I would like to thank Yu Fang, Sunming Qin, Nanjun Chen, Jipu Wang, Haining Zhou, Chen Hao, Xiaofan Cui, Fangyuan Chang, Xiaozhi Yu, Yumeng Ma, Wendi Ding, Duo Yang and Yifei Ma. Thanks to you, I could visit a lot of interesting places in and outside Michigan, have a lot of tasty food and know a lot of interesting stories. You guys open my eyes and help me get to know more colorful aspects of the world. And when I had emergencies, you would show up and help me get through

them. Your companionship, support and the time we spent together make my life much easier. To those that I have not explicitly mentioned here, please know that I feel so grateful for the moment we have shared.

I would especially like to thank Zhuo Chen, Tianshu Dong, Jin Li, Hongmei Tang and Haokui Xu. It is you who I play tennis with, talk with, take a walk with, explore new places with and go shopping with during this special time. The CVOID-19 changes my life, but you help me bring the life back to normal.

I also owe thanks to Xin Zan. You are more than a friend and an emergency contact. You are my sun.

Most importantly, I would like to thank my family. Every time we are chatting on the phone, you always talk a lot to remind me of taking good care of myself. Your words are simple, but I know your love is endless and is always there.

TABLE OF CONTENTS

ACKNOWLEDGMENTS . . . . **ii**

LIST OF TABLES . . . . **viii**

LIST OF FIGURES . . . . **x**

LIST OF ALGORITHMS . . . . **xiii**

LIST OF APPENDICES . . . . **xiv**

LIST OF ABBREVIATIONS . . . . **xv**

ABSTRACT . . . . **xviii**

CHAPTER

1 Introduction **1**

1.1 Motivation and Historical Review 1
1.1.1 Steady-state Methods 2
1.1.2 Transient Methods 3
1.2 Outline 5

2 Background **8**

2.1 Neutron Transport Equation 8
2.1.1 Time-dependent Neutron Transport Equation 8
2.1.2 Steady-State Neutron Transport Equation 10
2.1.3 Feedback in the Neutron Transport Equation 10
2.2 Overview of the Principle Methods 11
2.2.1 Monte Carlo Methods 11
2.2.2 Deterministic Methods 12
2.2.3 Common Approximations for Deterministic Codes 13
2.3 Steady-State Iteration Methods 17
2.3.1 Source Iteration 17
2.3.2 CMFD Acceleration 19
2.3.3 Methods for the Multigroup Diffusion Eigenvalue Problem 22
2.3.4 Other Solvers 26
2.3.5 Coupling with Other Physics Solver 26
2.3.6 Stability of Iteration schemes 28

2.4 Transient Methods in MPACT 29
2.4.1 Multigroup Transient Fixed Source Problem 29
2.4.2 Source Iteration and CMFD Acceleration 31
2.4.3 Exact Point Kinetics Equation 34
2.4.4 Transient Multilevel Scheme 36
2.4.5 Coupling with Other Physics 37
2.4.6 Flowchart of Transient Multilevel Method 38
2.4.7 Stability of Time-dependent NTE 38

3 Derivation of Nearly-optimally Partially Converged CMFD 40

3.1 Theory and Methodology 40
3.1.1 Model Problem 41
3.1.2 Nonlinear Diffusion Acceleration 42
3.1.3 Picard Iteration for NDA with Feedback 45
3.2 Fourier Analysis 45
3.2.1 Linearization 45
3.2.2 Fourier Ansatz 47
3.2.3 Fourier Analysis Result 48
3.2.4 Validation of Fourier Analysis Results 51
3.2.5 Effect of Partial Convergence 53
3.2.6 Partial Convergence vs Relaxation 54
3.3 Relaxation-free Iteration Scheme 57
3.3.1 NOPC Algorithm 57
3.3.2 Stability of NOPC-stabilized Method 61
3.3.3 Discussions of the Relaxation-free Method 63
3.4 Summary 65

4 Numerical Verification of Nearly-optimally Partially Converged CMFD 67

4.1 Introduction 67
4.2 Overview of Iteration Schemes 67
4.2.1 Feedback from Other Physics 67
4.2.2 Wielandt Shifted Power Iteration in Practice 68
4.2.3 CMFD Solvers in MPACT 69
4.3 Nearly-optimally Partially Converged CMFD Solver 70
4.3.1 Importance of the Partial Convergence of CMFD solutions 70
4.3.2 Estimation of the Nearly-optimal Partial Convergence 73
4.3.3 Estimation of the Localized Feedback Intensity 74
4.3.4 Verification of Feedback Intensity 78
4.3.5 Flowchart of the NOPC-CMFD 80
4.3.6 Summary 83
4.4 Numerical Results for Realistic Problems 84
4.4.1 3D Pin Problem 85
4.4.2 Single Assembly Problem 89
4.4.3 Effect of Iterative T/H Solutions 93
4.4.4 3×3 Core Problem 93

4.4.5 Full-core Single State Problem 100
4.4.6 Multistate Full-core Depletion Problem 102
4.5 Summary 103

5 Multiphysics Diffusion Acceleration with X-CMFD 105

5.1 Motivation and Model Problem 106
5.1.1 Motivation 106
5.1.2 Model Problem 106
5.1.3 The NFCDA Iteration 107
5.1.4 Iteration Procedure of X-CMFD 108
5.2 Fourier Analysis 109
5.2.1 Expression for Spectral Radius 109
5.2.2 Analysis Results 111
5.2.3 X-CMFD Iteration Scheme Design 114
5.2.4 X-CMFD vs NFCDA 117
5.3 Numerical Implementation 117
5.3.1 Prototype Implementation 118
5.3.2 Optimized Implementation 118
5.4 Numerical Results for Realistic Problems 120
5.4.1 Verification of Prototype 121
5.4.2 Single Assembly Problem 121
5.4.3 Affect of Iterative T/H Solutions 124
5.4.4 3×3 Core Problem 124
5.5 Summary 126

6 Transient Multilevel With One-group CMFD Acceleration 128

6.1 One Group CMFD Acceleration 128
6.1.1 1G/MGCMFD Iteration 129
6.1.2 1GCMFD Level in TML 132
6.2 Numerical Results 134
6.2.1 4-mini Test Problem 135
6.2.2 7×7 Test Problem 141
6.2.3 Efficiency of 1GCMFD Acceleration 144
6.2.4 Full-core Hot Full Power RIA Problem 146
6.3 Summary 151

7 Improvements of the Coupled Transient Simulation 153

7.1 Spectral Deferred Correction Method 153
7.1.1 Theory 155
7.1.2 SDC for PKE 162
7.1.3 SDC for PKE-EF 168
7.2 Improvement of CMFD Level 170
7.2.1 Decoupling TH inside the TML 170
7.2.2 New 1GCMFD Level Implementation 175
7.3 Operator Splitting Scheme 176

7.3.1 Evidence from PKE-EF Problem 177
7.3.2 Evidence from MPACT Simulations 179
7.4 Summary 181

8 Adaptive Time-stepping Methodology for Transient Multilevel 183

8.1 Existing ATS Methods 184
8.2 ATS Method for Neutronics Solver 185
8.2.1 Neutronics Internal Error Estimation 186
8.2.2 Step Size Calculation 189
8.2.3 Simple Numerical Results 191
8.3 ATS Method for Coupling 195
8.3.1 PKE-EF Verification 196
8.3.2 Coupling Error Estimation and Adaptive Methodology 199
8.3.3 Simple Numerical Results 203
8.4 ATS Flowchart 207
8.4.1 ATS Method for CMFD Level 208
8.4.2 Overall Flowchart 209
8.5 Performance Results 210
8.5.1 4-mini Test Problem 210
8.5.2 7×7 Test Problem 213
8.5.3 SPERT Test 86 Problem 216
8.6 Summary 218

9 Conclusion & Future Work 220

9.1 Conclusions 220
9.2 Future Work 222
9.2.1 Verification of Steady-state Iteration Scheme 222
9.2.2 Improvement on Transient Methodology 223

APPENDICES 225

BIBLIOGRAPHY 243

LIST OF TABLES

2.1 Definition of terms in time-dependent transport equations 9

3.1 Range of parameters used in Fourier analysis 49

4.1 Sensitivity study of the effect of γ_g in P6 (shielding enforced) 82
4.2 Sensitivity study of the effect of γ_g in P6 83
4.3 Convergence specification and relaxation for cases of the CMFD solvers 84
4.4 Convergence behavior of different iteration schemes in 3D P1 with different total powers 86
4.5 Run time of MPACT with different iteration schemes in 3D P1 with different total powers 87
4.6 Results of 3D P1 cases with feedback from TH 88
4.7 Results of 3D P1 cases with feedback from TC and TCE 89
4.8 Convergence behavior of different iteration schemes in P6 with different total powers . 90
4.9 Run time comparison of different CMFD methods in P6 with different total powers . . 91
4.10 Convergence behavior of different iteration schemes in P6 with different total powers with CTF as TH Solver 94
4.11 Specification of the P4 cases 94
4.12 Convergence behavior of iteration schemes in P4 cases with different control rod positions (20% rated power) 96
4.13 Run time comparison of iteration schemes in P4 cases with different control rod positions (20% rated power) 96
4.14 Convergence behavior of iteration schemes in P4 cases with different control rod positions (100% rated power) 98
4.15 Run time comparison of iteration schemes in P4 cases with different control rod positions (100% rated power) 99
4.16 Convergence properties of the MPACT with different iteration schemes in P7 101
4.17 Convergence properties of the MPACT with different iteration schemes in P9 102

5.1 Number of outer iterations to converge 3D P1 with feedback from TH 121
5.2 Number of outer iterations to converge P6 with 140% rated power 122
5.3 Convergence comparison of X-CMFD solvers to other CMFD solvers for P6 123
5.4 Number of outer iterations to converge P6 for X-CMFD (shielding calculation enforced)124
5.5 Number of outer iterations to converge P6 for MSPLC with CTF as the TH solver . . . 124
5.6 Convergence comparison of the X-CMFD solvers to other CMFD solvers for P4 cases at 100% rated power 125

5.7 Convergence comparison of the X-CMFD solvers to other CMFD solvers for P4 cases at 20% rated power. 126

6.1 TML options for seven different cases based on the 4-mini test problem 137
6.2 Run time for different cases based on the 4-mini test problem 137
6.3 L_1 and L_∞ norms of the relative error for the core power solutions of the 4-mini test problem 141
6.4 L_1 and L_∞ norms of the relative error for the core power and reactivity solution of the 7×7 test problem 141
6.5 Parameters for five different cases based on the 7×7 test problem 143
6.6 Run time for different cases based on the 7×7 test problem 143
6.7 Average number of GMRES iterations to solve the MGCMFD TFSP system 145
6.8 Summary of run time and iteration number for solving an MGCMFD TFSP system . . 145
6.9 Time step sizes for the full-core HFP RIA problem 148
6.10 Run time for different cases of the full-core HFP RIA problem 151
6.11 Comparison of k_{eff} for the full-core HFP RIA problem 151

7.1 Delayed parameters of a PKE problem 154
7.2 Accuracy comparison of approximation 2 to approximation 3 for various problems . . 173
7.3 Run time comparison of approximation 2 to approximation 3 174

8.1 Comparison of the peak power times estimated by the PKE-EF model and calculated by MPACT 199
8.2 ATS cases performed in the SPERT-2D problem 204
8.3 Accuracy for different CMFD steps per transport step in the 4-mini test problem . . . 208
8.4 Tolerances for cases with the ATS methods in the 4-mini test problem 211
8.5 Comparison of adaptive simulations to the simulations with default step size for the 4-mini test problem 211
8.6 Settings for cases with adaptive methods in the 7×7 problems 214
8.7 Comparison of the ATS simulations to the simulations with default step size for the 7×7 test problem 214
8.8 Comparison of adaptive simulations to the simulations with default step size for the SPERT test 86 problem 217

LIST OF FIGURES

2.1 Flowchart of multiphysics scheme for steady-state calculation in MPACT 27
2.2 Illustration of the TML Scheme 36
2.3 Flowchart for TML coupled with TH feedback 39

3.1 Comparison between Fourier analysis and numerical results 52
3.2 Spectral radius as a function of power iteration number L 53
3.3 Spectral radius as a function of the WS ratio r 53
3.4 Relaxation factor $\beta(\omega)$ and iterative eigenvalue $|\theta(\omega)|$ as functions of Fourier frequency 56
3.5 Maximum spectral radius as a function the ω_p used to estimate the NOPC 59
3.6 Verification of the ω_p to estimate NOPC 60
3.7 Stability of the nearly-optimally partially converged odCMFD 62
3.8 Stability of the nearly-optimally partially converged lpCMFD 62
3.9 Spectral radius of NDA with Algorithm 9 in problems with various combinations of c and γ 63

4.1 The effect of the number of WS power iterations on convergence of MPACT in P6. . . 71
4.2 Convergence behavior of MPACT with varying relaxation factor 72
4.3 Illustration of the Red-Black perturbation 74
4.4 Flowchart of feedback intensity calculation 76
4.5 Estimated γ_m as a function of different perturbation factors α 77
4.6 Estimated $\bar{\gamma}$ as a function of different perturbation factors α 78
4.7 The relation between the characteristic feedback intensity and the power 79
4.8 Flowchart of NOPC-CMFD iteration scheme 80
4.9 Evolution of the fission source residual with the outer iteration for 3D P1 cases 85
4.10 Comparison of NOPC-CMFD solvers to other CMFD methods for the convergence in the 3D P1 cases 88
4.11 Variation of the fission source residual with the outer iteration for P6 cases 91
4.12 Problem 4 assembly, poison and control layout 95
4.13 An example of the axial power profile of P4 95
4.14 Comparison of NOPC-CMFD solvers to other CMFD methods for P4 cases 97
4.15 Assembly and poison layout of the full-core problem (quarter symmetry) 100
4.16 Evolution of the $k-$eigenvalue residual and the fission source residual for P7 101

5.1 Stability of the NDA with fully-coupled low-order problem 111
5.2 Comparison of Fourier analysis and numerical results of X-CMFD 112
5.3 Spectral radius as a function of the problem optical thickness for different M 113

5.4 Spectral Radius vs scattering ratio and power iteration number 114
5.5 Spectral radius as a function of scattering ratio and the number of inner iterations for different γ 115
5.6 Spectral radius as a function of scattering ratio and the number of inner iterations for different WS ratios 116
5.7 Flowchart of X-CMFD iteration scheme in MPACT 118
5.8 Implementation for the solution of X-CMFD equation 119

6.1 Flowchart for 1G/MGCMFD iteration 131
6.2 Flowchart of dynamic 1G/MGCMFD iteration strategy 132
6.3 TML with one-group CMFD level 133
6.4 Flowchart of TML iteration scheme with 1GCMFD acceleration 134
6.5 Assembly layout and geometry of the 4-mini test problem 136
6.6 Power history for different cases of the 4-min problem. 138
6.7 Error of the power results of different 4-mini test problem cases. 138
6.8 Comparison for the core power of the 4-mini test problem with different methods . . . 140
6.9 Geometry of the 7×7 test problem 142
6.10 Power and reactivity histories for the 7×7 test problem 143
6.11 Relative errors of power history and reactivity history for the 7×7 test problem 144
6.12 Evolution of fission source for different CMFD solution methods. 145
6.13 Assembly and poison layout of the full-core problem (quarter symmetry) 147
6.14 Control rod layout of the full-core RIA problem 148
6.15 Power history of the full-core HFP RIA problem 149
6.16 Reactivity history and comparison of the full-core HFP RIA problem 150
6.17 Core power and maximum linear rate comparison for the full-core HFP RIA problem . 150

7.1 Accuracy of BE method in various problems 154
7.2 Stability region of SDC 162
7.3 Effect of the endpoint calculation on order of convergence. 166
7.4 Effect of low-order solvers for SDC on order of convergence. 166
7.5 Order of convergence of the SDC method for the PKEs 167
7.6 Comparison of results for EPKE solvers based on BE to SDC reference in MPACT . . 168
7.7 Convergence study of a PKE-EF solver based on SDC 170
7.8 Illustration of TH coupling for TML 170
7.9 Comparison of approximations used for the CMFD coefficients calculation in TML . . 172
7.10 Illustration of deficiency of approximation 3 172
7.11 Error evolution of approximation 2 and approximation 3 for the SPERT-3D-HZP case 174
7.12 TML-4 with the new implementation of 1GCMFD level 176
7.13 Multiphysics coupling scheme for transient simulation used in MPACT 177
7.14 Inconsistency between error and time step size for LS in PKE-EF problems 178
7.15 Error for varying time steps in SS for PKE-EF problem 180
7.16 Comparison of LS and SS with varying time step sizes 180
7.17 Comparison of LS and SS with constant time step sizes 181

8.1 Investigation of characterization of the neutronics internal error 187

8.2 Ratio of the multi-step error to the one-step error for a pin problem 189
8.3 Performance of the adaptive methodology for controlling the internal error of TML . . 192
8.4 Evolution of the power growth ratio with the adaptive method 193
8.5 Performance of ATS method for controlling the internal error in the 4-mini test problem 195
8.6 Relative error and the time step size for different internal error tolerances 196
8.7 Flowchart of adaptive methodology for coupling 197
8.8 Evolution of linear energy feedback coefficient 198
8.9 Comparison of one-step error and multi-step error 200
8.10 Evolution of the relative difference as a function of time 201
8.11 Evolution of the power as a function of time for the SPERT-2D cases 203
8.12 Relative error of the power and time step size as a function of simulation time in the SPERT-2D-HZP problem 204
8.13 Relative error of the power and time step size as a function of simulation time in the SPERT-2D-HFP problem 206
8.14 Comparison of the performance of the ATS method in the SPERT-2D-HZP cases and the PKE-EF model 207
8.15 Accuracy of adaptive CMFD level in the 4-mini test problem 209
8.16 Flowchart of the ATS methods for TML 210
8.17 Relative errors as a function of time in the 4-mini test problems 211
8.18 Time step size as a function of time in the 4-mini test problems 212
8.19 Time step size as a function of time in the 7×7 test problem 215
8.20 Comparison of the relative error of the power for different cases of the 7×7 problem. . 215
8.21 Measured results and the simulated results for the SPERT test 86 problem. 216
8.22 Comparison of the relative error of the power and time step size for different cases of the SPERT test 86 problem. 217

B.1 Layout of the SPERT-2D test problem 241
B.2 Layout of the SPERT-3D test problem 242

LIST OF ALGORITHMS

1 Source iteration scheme for k-eigenvalue neutron transport problem 18

2 CMFD acceleration for k-eigenvalue problem 22

3 Power iteration algorithm 23

4 Wielandt shifted power iteration algorithm 24

5 Simplified algorithm of MSED solver 25

6 Source iteration scheme for TFSP 32

7 CMFD acceleration for TFSP 33

8 NDA iteration scheme for the neutron transport problem with feedback. 44

9 Nearly-optimal partial convergence estimation 61

10 Perturbing the flux in MPACT for calculating feedback intensity 75

11 Fully coupled NFCDA for k-eigenvalue problem with feedback 107

12 X-CMFD for k-eigenvalue problem with feedback 108

13 General Procedure for Spectral Deferred Correction 160

14 Spectral Deferred Correction for PKEs 165

15 Estimation of the errors to be controlled in TML. 188

16 Time step size for control the neutronics internal error. 192

17 Estimation of the errors to be controlled for coupling. 200

LIST OF APPENDICES

A Fourier Analysis of the Stability of the Steady-state iteration Methods **225**

B Description of Transient Problems **240**

LIST OF ABBREVIATIONS

AA Anderson acceleration

AIC silver-indium-cadmium

ATS adaptive time-stepping

BE backward Euler

BWR boiling water reactor

CASL Consortium for Advanced Simulation of Light Water Reactors

CB critical boron search

CFD computational fluid dynamics

CMFD Coarse Mesh Finite Difference

CTF COBRA-TF

EPKE exact point-kinetics equation

EX equilibrium Xenon

FP first-order precursor integration

HZP hot zero power

HFP hot full power

IQS Improved Quasi-static

IMEX Implicit-Explicit

JFNK Jacobian-free Newton-Krylov

LWR Light Water Reactor

LS Lie Splitting

MAF Multigrid Amplitude Function

MC Monte Carlo

MGCMFD Multigroup CMFD

MGPI Multigroup CMFD solver with power iteration

MOC Method of Characteristics

MPACT Michigan Parallel Characteristics Transport

MSED Multilevel-in-Space-and-Energy Diffusion

NDA Nonlinear Diffusion Acceleration

NEM Nodal Expansion Method

NFCDA Nonlinear Fully Coupled Diffusion Acceleration

NOPC nearly-optimal partial convergence

NOPC-CMFD nearly-optimally partially converged CMFD

NTE neutron transport equation

PARCS Purdue Advanced Reactor Core Simulator

PCQM Predictor-Corrector Quasi-Static method

PI Power Iteration

PKE point-kinetics equation

PKE-EF PKE with linear energy feedback

PWR pressurized water reactor

RHS right-hand side

RIA reactivity insertion accident

odCMFD Optimally Diffusive CMFD

OS operator splitting

SPERT Special Power Excursion Reactor Test

SP second-order precursor integration

SDC Spectral Deferred Correction

SS Strang Splitting

SMR small modular reactor

RCCA rod cluster control assembly

TCP$_0$ Transport-Corrected P_0

TFSP Transient Fixed Source Problem

TH Thermal Hydraulics

TML Transient Multilevel

TML-3 original TML

TML-4 TML with one-group CMFD level

WS Wielandt shifted

XS Cross Section

1GCMFD one-group CMFD

3D three-dimensional

ABSTRACT

Modeling the time-dependent transient behavior of nuclear reactors with high-fidelity pin-resolved detail has increased importance when the operating power of the reactor is increased to improve the economic performance. In previous research, the efficiency of the solution of the steady-state neutron transport equation (NTE), which provides the initial condition for the transient, was improved with the development of advanced methods such as the Multilevel-in-Space-and-Energy Diffusion (MSED). However, the application of the MSED method was ultimately limited by numerical instabilities in the presence of cross section feedback. The first objective of this research is to improve the efficiency of the steady-state solution by investigating and eliminating the numerical instability of accelerated neutron transport iterations when there is cross section feedback.

The second objective of the research here is to address the computational costs of performing transient simulations by improving the performance of the Transient Multilevel (TML) method in the MPACT code. Specifically, the run time of the Coarse Mesh Finite Difference (CMFD) solver in TML dominates the run time, so a one-group acceleration method is developed and added. Automated time-stepping methods were also not previously available for TML. The research here significantly improves the efficiency of the transient calculation by accelerating the CMFD solver and using adaptive time-stepping methods. Improving the stability and efficiency of the transient whole-core neutron transport calculations is the main significant and original contribution of this work.

The specific contributions of this thesis for the steady-state calculation are the theory, development, and implementation of the nearly-optimally partially converged CMFD (NOPC-CMFD) method and the X-CMFD method in MPACT. As its name suggests, the NOPC-CMFD method stabilizes the iteration scheme by determining and utilizing the nearly-optimal partial convergence of the diffusion solutions. The X-CMFD method is an original method that stabilizes the iteration by applying the feedback at the power iteration level of the low-order diffusion eigenvalue problem. Compared to the default iteration scheme in MPACT, the methods developed here demonstrate the same stability compared to CMFD-accelerated transport iterations in problems without feedback, and reduce the overall run time of the full-core multi-state depletion problem by $\sim$43%.

The principal original work of this thesis for the transient simulations is the introduction of a one-group CMFD (1GCMFD) acceleration method and the development of adaptive time-stepping methods to further accelerate the TML scheme. The 1GCMFD method is shown to reduce the overall computational time of CMFD by as much as 50% for practical large-scale applications. The adaptive time-stepping method introduced adjusts the time step so that the maximum magnitude of the relative error is smaller than 1% for the applications considered in this research. Other innovative methods include the usage of the Spectral Deferred Correction (SDC) method to solve the point-kinetics equation and the use of Strang Splitting (SS) to replace Lie Splitting for coupling the neutronics and the TH solvers. The implemented SDC method is A-stable for orders up to 8, and the SS addresses the inconsistency between the error and time step size when the time step size is varied.

When the 1GCMFD acceleration and the adaptive methods are applied together, the performance of the TML scheme for the Special Power Excursion Reactor Test (SPERT) test 86 problem is reduced by $\sim$22% and the maximum magnitude of the relative error is reduced from $\sim$1.8% to $\sim$0.4%, compared to the use of TML with the default parameters.

CHAPTER 1

Introduction

1.1 Motivation and Historical Review

Accurate and high-fidelity modeling of nuclear reactors by solving the whole-core neutron transport problem has been the state-of-the-art for nuclear reactor simulation for over a decade [1, 2, 3, 4, 5]. Programs such as Consortium for Advanced Simulation of Light Water Reactors (CASL) and Nuclear Energy Advanced Modeling and Simulation (NEAMS) have supported the development of advanced methods in the high-fidelity simulations. The high-fidelity simulations are important because they can help provide insight into several complex phenomena in nuclear reactor analysis.

A principle focus is modeling the transient behavior of a nuclear reactor with pin-resolved detail. The basis is to solve the Boltzmann neutron transport equation (NTE) directly without spatial homogenization and low-order approximations such as diffusion theory [6]. The procedure to perform transient simulation consists of two parts. The first part is the solution of the steady-state NTE that is an eigenvalue problem. The steady-state neutron transport calculation provides the initial condition for the second part, that is the solution of the time-dependent NTE.

The CASL and NEAMS programs advance the state-of-the-art of the high-fidelity multiphysics simulation significantly [4, 7], However, the neutron transport simulation is computationally expensive due to its high-dimensionality. Furthermore, there are considerable challenges with the feedback from other physics that is always present in power reactors. Therefore, the issues that the iterative method for solving the steady-state NTE is not robust and that it can be formidable to maintain suitable accuracy while minimizing computational resources for realistic transient problems still exist.

The lack of robustness of the current iteration scheme in the steady-state calculation, and the relatively long run time for the transient calculation are the motivations for the work in this thesis. This thesis work is comprised of two topics. One focuses on improving the robustness of the steady-state iterative methods. The other focuses on improving the efficiency of the transient methods.

1.1.1 Steady-state Methods

Historically, the whole-core neutron transport problem without feedback has been well studied with both deterministic codes [8, 4], and stochastic codes [9, 10]. Successful acceleration methods have been developed to make the solution of the whole-core high-fidelity neutronics problem tractable. One widely-used technique is the Coarse Mesh Finite Difference (CMFD) method [11, 12], which can be understood as a generalization of Nonlinear Diffusion Acceleration (NDA) [13] that also coarsens on the spatial grid. NDA, and its variants, utilize the solution of the low-order diffusion calculation to "precondition" (or synthetically accelerate) the high-order transport solution. Considerable research has been performed to improve the convergence rate, stability and efficiency of CMFD over the past decade [14, 15, 16, 17, 18].

However, nearly all of this work has not considered the multiphysics problem in the theoretical formulation and analysis of the NDA/CMFD solvers. The development of the iteration scheme for modeling the multiphysics problem is always treated as a completely separated task.

For the multiphysics simulation, there have been numerous research efforts that have attempted to improve the robustness and performance–primarily through numerical experimentation [19, 20, 21, 22]. Most of these works have relied on the use of fixed-point iteration schemes (e.g. Picard iteration) [3, 4, 23, 24, 25, 26, 27, 28, 29, 30]. The wide use of the Picard iteration approach is due to its simplicity of implementation, and also results from the context that the software for simulating the different reactor physics usually comes from separate, stand-alone codes. In this type of "code-coupling", the convergence of the steady-state problem is generally assumed to be achieved when the solutions of the different solvers stop changing (i.e. a fixed point is reached).

In nearly all implementations of the Picard iteration in this context, issues such as the slow rate of convergence or the lack of robustness have been observed. This can be shown for some simple numerical cases [26, 31] and has also been reported for more realistic problems [32, 23, 33, 34]. In these previous works, there are two basic approaches used to address the issues of stability and robustness; these approaches involve implementing under-relaxation or partially converging each physics[22, 35]. Sometimes these two strategies are used in tandem. The motivation for using relaxation is grounded in that (i) this is a simple thing to try when the iterations become oscillatory, and (ii) it usually works. The logic for only partially converging the iterations is intuitive in the sense that the method should not "waste time" converging an intermediate iterate of the transport equation where we know the coefficients are not well converged. These approaches will not necessarily work for every problem, and therefore they are not ideal.

Methods such as the Jacobian-free Newton-Krylov (JFNK) method [36, 34] and the Anderson acceleration (AA) method [37, 38] are more stable than Picard iteration, and have been introduced into the nuclear reactor simulation. These methods are proposed mathematically to couple equations of different physical fields. The JFNK method requires a significant change in the codes involved. Therefore, it is not suitable for most of the code-coupling systems such as the Virtual Environment for Reactor Analysis (VERA) supported by CASL [4]. The AA method is proposed to mix the solutions of previous iterations adaptively. It is much easier to implement compared to the JFNK method. Very recent research [38] has shown it helps to achieve much better stability and efficiency for the CMFD-accelerated transport scheme. Since the AA method can be understood as a method with more complicated relaxation factors, its performance may be limited as suggested by [39].

Nevertheless, the coupling of the feedback from other physics with the CMFD-accelerated neutron transport is *the Picard method implemented by most high-fidelity neutronics codes*, including DeCART [40], nTRACER [3], STREAM [41], and Michigan Parallel Characteristics Transport (MPACT) [4]. Within the multiphysics iteration, the transport problem is solved with one CMFD-accelerated "outer" transport iteration, and then relaxation is applied (typically to the power, but potentially the temperature or density) when exchanging the data with a different feedback solver.

For this Picard iteration scheme, an important and perhaps counter-intuitive observation made recently with respect to the use of CMFD acceleration, is that when one more tightly converges the low-order diffusion solutions, the coupled iteration scheme converges more slowly and becomes less stable. As better acceleration methods for neutronics problems, such as the Multilevel-in-Space-and-Energy Diffusion (MSED) method, are being developed for problems without feedback, the tight convergence of the low-order diffusion solutions creates the issue of having a less robust multiphysics iteration scheme. This ultimately leads to having to impose a penalty on the convergence rate (through relaxation) of the acceleration scheme [32].

The first contribution of this thesis is the work in developing novel methods to improve the robustness of the whole-core steady-state simulations. Prior to this work, the slow convergence rate of the advanced acceleration method in problems with feedback was not fully understood. This work seeks to analyze the stability of the CMFD-accelerated transport scheme in problems with feedback and propose methods that let the iteration scheme have an almost constant convergence rate irrespective of the presence of the feedback.

1.1.2 Transient Methods

Historically, researchers have performed a considerable amount of research to solve the time-dependent NTE directly [40, 42, 43, 44]. However, direct transient simulation without acceleration requires thousands of time steps for simulating a typical super-prompt critical transient. Therefore, novel methods have been investigated to speed up the transient simulation. One of the recently successful pin-resolved transient transport methods [45, 46, 47] developed is Transient

Multilevel (TML). This method has been recently implemented into MPACT to accelerate the transient transport calculation. The method makes it feasible to run three-dimensional (3D), full-core time-dependent simulations with pin-resolved details for nuclear reactors in a reasonable time [48, 49, 50, 51].

The TML scheme is a multilevel algorithm using three levels with 3D-transport coupled to 3D-CMFD and 3D-CMFD coupled to the exact point-kinetics equations (EPKEs) to effectively capture the evolution of the flux. In each level, the flux solution is estimated with a predictor with coarse time step size and is factorized as a shape function and an amplitude function. The shape function is assumed to vary more slowly than the amplitude function in time, while the amplitude function is solved with a fine time-step corrector. The TML scheme is a method based on the ideas of Improved Quasi-static (IQS) method in [52] and the Multigrid Amplitude Function (MAF) method in [45], while employing the Predictor-Corrector Quasi-Static method (PCQM) [53] on both the transport/CMFD coupling and CMFD/EPKEs coupling.

The TML scheme has been shown to increase the efficiency of running the transient transport simulation without degradation of accuracy [54]. It has been shown that the TML scheme reduces the total computation time by around 80% compared with the pure 3D-transport transient scheme with averaged numerical relative error less than 1% in for Special Power Excursion Reactor Test (SPERT) problem [51].

However, even with the advancements of TML, the total calculation time is still quite large for some problems. The total calculation time for the SPERT [55, 56] test 86 case with 2880 cores on the Titan compute cluster at Oak Ridge National Laboratory (ORNL) was around 2 hours [51]. One of the most important contributions to the long run times is that the CMFD solver run time dominates the total calculation time. Therefore, it is a key component to improve to speed up the overall calculation. There are two reasons why the CMFD solver takes a relatively large portion of the computational effort (i) the Multigroup CMFD (MGCMFD) system constructed during each time step is nearly singular or ill-conditioned and therefore IS hard to solve; (ii) the number of CMFD calculations per time step is large to capture the changes of the cell-wise amplitude functions; (iii) the parallel CMFD calculation is inefficient.

An additional challenge is that when the TML is used for the transient simulation, no strategy is used to determine the time step size and the number of CMFD calculations. For most of the simulations, the time step size is set manually, and the number of CMFD calculations on the CMFD level is a constant for each time step. What can be expected for this case is that the time step size can be too coarse to capture the variation of the fast-changing physical fields, and be too fine to achieve optimal accuracy when the solution varies slowly. Therefore, the current practice of TML still has some notable inefficiencies in trying to balance the accuracy and computational cost.

The second contribution of this thesis is the work in developing novel methods to improve the efficiency of the TML [54] scheme. This work seeks to reduce the run time of CMFD solver and develop the adaptive time-stepping (ATS) methods for TML to further improve the efficiency of the transient calculation.

The overarching goal of this research is to develop a set of methods so that practical 3D whole-core time-dependent neutron transport simulation coupled with multiphysics is robust and efficient.

1.2 Outline

The remainder of this thesis is structured as follows.

In Chapter 2, a general introduction of the theory and numerical methods for performing neutron transport simulations is provided. Starting from the time-dependent continuous neutron transport problem, various approximations are applied to obtain the formulations for the problems of interest: the multigroup steady-state neutron transport equation and the multigroup time-dependent neutron transport equation. The state-of-the-art numerical methods, and the important details for solving these equations are described to the extent required to understand the work in this thesis. These methods include the CMFD acceleration, iteration schemes to solve the steady-state NTE coupled with feedback from other physics, the transient methodology for time-dependent NTE, and the methods to couple the neutronics and feedback. No new theory is developed in this chapter.

Chapter 3 focuses on the analysis of the steady-state eigenvalue problem that is the initial condition for the transient simulation. Specifically, a Fourier analysis for a simplified CMFD-accelerated neutron transport problem with feedback from flux-dependent cross sections is developed to provide theoretical insight into observable numerical instabilities in the steady-state neutron transport calculation. The Fourier analysis result is used to derive the relationship between relaxation and the partial convergence of the CMFD accelerated transport. Using this relationship a new and more robust relaxation-free multiphysics iteration scheme is developed. This method eliminates the need for a relaxation factor by developing equations to nearly-optimally partially converge the CMFD equations used to iteratively accelerate the convergence of the multiphysics transport sweep. This new method is called the nearly-optimally partially converged CMFD (NOPC-CMFD) method. It is one of the significant and original contributions in this thesis.

The implementation of the new method developed in Chapter 3 is applied to various test cases that are presented in Chapter 4. Chapter 4 focuses on developing a methodology for estimating "a key problem-dependent parameter", the feedback intensity, required by the NOPC-CMFD method. Next, the implementation of NOPC-CMFD in MPACT is described and several numerical calculations are performed. Problems ranging from a single pressurized water reactor (PWR) fuel rod to a full-core PWR cycle depletion are analyzed to assess the performance and robustness of NOPC-CMFD over a wide range of conditions that consider multiple forms of multiphysics feedback.

The conclusions of Chapter 3 suggest a number of ways to achieve robustness. The NOPC-CMFD method is the first one and modifies the existing iterative methods. Chapter 5 presents a more fundamental rethinking of the state-of-the-art algorithms and considers the problem of feedback from the beginning. The results show that the current iteration scheme can have the largest region of stability when the multiphysics feedback is applied at the power iteration of the low-order eigenvalue problem. The new method is called the X-CMFD. The key feature of the X-CMFD method is that the more tightly the low-order diffusion solution is converged, the more stable the iteration scheme is. Therefore, sophisticatedly determining the partial convergence as NOPC-CMFD does can be avoided. The X-CMFD method is also implemented in MPACT and optimized for the practical simulations. Numerical results that range from a single PWR fuel rod to a PWR 3×3 core are presented.

Chapter 6 shifts the focus to the transient problem. The one-group CMFD (1GCMFD) system is introduced to accelerate the transient calculation. 1GCMFD primarily accelerates the convergence of the source for MGCMFD calculations through 1G/MGCMFD iteration. And a new 1GCMFD level is introduced into the TML method to alleviate the computation expense of MGCMFD. The new TML with four computational levels is referred to as TML with one-group CMFD level (TML-4). Various numerical cases are used to investigate the practicality of the 1G/MGCMFD iteration and TML-4 scheme. Numerical results show that using 1G/MGCMFD iteration with the dynamic iteration strategy alone does better to capture the evolution of the amplitude function when the scalar flux distribution in energy space varies rapidly, and thus provides more accurate results. However, TML-4 is only more efficient in capturing the variation of the energy-integrated amplitude when cross section changes are small, and feedback dominates the change of the reactivity. It does not help significantly when there are large Cross Section (XS) perturbations like when a control rod is ejected. Therefore, the TML-4 scheme is limited in that it does not in general improve the overall accuracy of the transport calculation using TML, and only addresses the computational inefficiency of the CMFD level.

This motivates some simple but effective modifications to improve the overall accuracy of the transient calculation with the TML scheme. These modifications are the focuses of Chapter 7. First is the adoption of the Spectral Deferred Correction (SDC) method that is introduced to replace the backward Euler (BE) in the solution of the EPKEs. It is shown that the SDC method is A-stable [57] for orders up to 8. For practical simulations, it is found that the SDC solver could let the TML scheme produce comparably accurate results compared to the TML scheme with ultra-fine EPKE steps. Next, the TML scheme is decoupled from the multiphysics in the CMFD level.

This modification improves the accuracy of the TML scheme in problems with control rod movement, and helps develop a new implementation of the 1GCMFD level for TML-4. The new TML-4 is more accurate compared to what was presented in Chapter 6. Lastly, the operator splitting (OS) to couple the neutronics solver and the Thermal Hydraulics (TH) solver is changed from Lie Splitting (LS) to Strang Splitting (SS). It is observed that LS would produce inconsistent behavior between the error and the time step size. In practical simulations, SS has comparable efficiency to LS, but produces more accurate results.

Finally, in Chapter 8, the ATS methods are developed for the TML scheme in problems with the feedback. Two adaptive methods are developed separately to limit the errors from the neutronics solution and the OS. An additional method to determine the number of CMFD steps in the CMFD level is also proposed. The two ATS methods are derived based on point-kinetics equation (PKE) models. The PKE models are used to estimate the temporal error and predict the maximum error in a time range. The time step size is calculated so that all the errors estimated by the two models are smaller than the user-specified error tolerances, and the number of CMFD steps is adjusted accordingly. Numerical results indicate that the current default time step size for time-dependent simulation is neither fine enough to capture the variation of flux when reactivity is being inserted, nor efficient after the reactivity insertion. Compared to the old scheme, the ATS methods generally improve the accuracy at the cost of efficiency for stages of reactivity insertion, and improve the efficiency after the reactivity insertion.

Finally, Chapter 9 provides conclusions and areas of future work.

CHAPTER 2

Background

This chapter provides the theoretical and numerical backgrounds needed to understand the work in this thesis. The chapter starts with the introduction of the continuous neutron transport equation (NTE) in Section 2.1. Forms of both the steady-state and the time-dependent NTE are briefly introduced. They are the fundamental equations for high-fidelity nuclear reactor simulations. Section 2.2 describes the basic methods to solve the NTE, and introduces the discretization and approximations made to arrive at the multigroup NTE suitable for numerical computation. The methods to solve the steady-state NTE are introduced in Section 2.3, and the methods for the time-dependent NTE are introduced in Section 2.4.

2.1 Neutron Transport Equation

In this section, both the time-dependent and the steady-state k-eigenvalue neutron transport equation are presented.

2.1.1 Time-dependent Neutron Transport Equation

The high-fidelity time-dependent transport simulation must solve the 3D time-dependent Boltzmann neutron transport equations and delayed neutron precursor equations given by

$$\begin{aligned}\frac{1}{v(E)}\frac{\partial\varphi\left(\boldsymbol{r},\boldsymbol{\Omega},E,t\right)}{\partial t}&+\Sigma_t(\boldsymbol{r},E,t)\varphi\left(\boldsymbol{r},\boldsymbol{\Omega},E,t\right)+\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\varphi\left(\boldsymbol{r},\boldsymbol{\Omega},E,t\right)\\&=\int_0^\infty\int_0^{4\pi}\Sigma_s\left(\boldsymbol{r},\boldsymbol{\Omega}\cdot\boldsymbol{\Omega}',E'\to E,t\right)\varphi(\boldsymbol{r},\boldsymbol{\Omega}',E',t)d\boldsymbol{\Omega}'dE'\\&+\frac{1}{4\pi}\chi_p(\boldsymbol{r},E,t)\Big[1-\beta\left(\boldsymbol{r},t\right)\Big]S_F\left(\boldsymbol{r},t\right)+\frac{\chi_d(\boldsymbol{r},E,t)S_d\left(\boldsymbol{r},t\right)}{4\pi}\,,\end{aligned}\tag{2.1}$$

and

$$\frac{dC_k\left(\boldsymbol{r},t\right)}{dt}=\beta_k\left(\boldsymbol{r},t\right)S_F\left(\boldsymbol{r},t\right)-\lambda_k\left(\boldsymbol{r},t\right)C_k\left(\boldsymbol{r},t\right)\ ,\ k=1,2,\dots\ K\,.\tag{2.2}$$

S_F and S_d are the total fission neutron source and delayed neutron source, respectively. These terms are defined as

$$S_F(\boldsymbol{r},t) = \frac{1}{k^0_{eff}} \int_0^\infty \nu(E')\Sigma_f(\boldsymbol{r},E',t)\phi(\mathbf{r},E',t)dE' \,, \tag{2.3}$$

and

$$S_d(\boldsymbol{r},t) = \sum_{k=1}^{K} \lambda_k(\boldsymbol{r},t)C_k(\boldsymbol{r},t)\,. \tag{2.4}$$

The initial eigenvalue k^0_{eff} of the steady-state NTE is included in the fission source term to ensure the criticality of the initial state. k^0_{eff} is obtained by solving the steady-state NTE, that is discussed in Section 2.1.2. The independent variables are t, $\boldsymbol{r}$, $\boldsymbol{\Omega}$ and E, which form the phase-space of the solution. The definition of the terms in Eqs. (2.1) to (2.4) are summarized in Table 2.1. It should be noted that β and χ_d are typically obtained by fission-source weighting, since delayed neutron fractions and spectra are different for different fissionable isotopes.

The term of the greatest interest for the time-dependent simulation is the neutron scalar flux, defined as

$$\phi(t,\boldsymbol{r},E) \equiv \int_{4\pi} \varphi(t,\boldsymbol{r},E,\boldsymbol{\Omega})d\boldsymbol{\Omega}\,, \tag{2.5}$$

where the angular flux, φ, is the fundamental unknown of the NTE.

An accurate pin-resolved scalar flux result is very important for the analysis of the nuclear reactor. It can provide insight into several complex phenomena on the sub-pin level to researchers, and it is used by designers in nuclear reactor safety analysis.

Table 2.1: Definition of terms in time-dependent transport equations.

Term	Definition	Term	Definition
φ	Angular flux	ϕ	Scalar flux
$\boldsymbol{\Omega}$	Direction of neutron flight	E	Energy
$\boldsymbol{r}$	Position	v	Velocity
Σ_t	Total cross section	Σ_s	Scatter cross section
Σ_f	Fission cross section	ν	Fission neutron yield
C	Delayed neutron precursor density	k	Precursor group index
χ_p	Prompt fission neutron spectrum	χ_d	Delayed fission neutron spectrum
β	Delayed neutron fraction	λ	Precursor decay constant
S_F	Prompt fission neutron source	S_d	Delayed fission neutron source
K	the maximum number of precursor groups.		

2.1.2 Steady-State Neutron Transport Equation

The transient simulation relies on the initial condition which must be obtained via the steady-state neutron transport calculation, where the time derivative terms are neglected. Therefore, Eq. (2.1) is simplified as:

$$\begin{aligned}\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\varphi\left(\boldsymbol{r},\boldsymbol{\Omega},E\right)+\Sigma_t(\boldsymbol{r},E)\varphi\left(\boldsymbol{r},\boldsymbol{\Omega},E\right)=\frac{\chi(\boldsymbol{r},E)}{4\pi k_{eff}}\int_0^{\infty}\nu\left(()\,E'\right)\Sigma_f\left(\boldsymbol{r},E'\right)\phi\left(\boldsymbol{r},E'\right)dE'+\\ \int_0^{\infty}\int_0^{4\pi}\Sigma_s\left(\boldsymbol{r},\boldsymbol{\Omega}\cdot\boldsymbol{\Omega}',E'\to E\right)\varphi(\boldsymbol{r},\boldsymbol{\Omega},E')d\boldsymbol{\Omega}'dE'\,.\end{aligned} \tag{2.6}$$

This is the steady-state neutron transport equation, and is a generalized eigenvalue problem. χ is the averaged fission spectrum weighted by prompt, χ_p, and delayed fission neutron, χ_d, spectrums. The quantities of interest are the eigenvalue, k_{eff}, and the corresponding eigenvector, ϕ. The k_{eff}, which is also called as multiplication factor in the nuclear reactor physics field, indicates the criticality of the system. If k_{eff} is **one**, then the system is critical and will remain at the current conditions unless otherwise changed. A k_{eff} less than one indicates that the system is subcritical, and that the reactor system is unable to sustain the chain reaction of nuclear fission reactions to produce power. Finally, a k_{eff} greater than one indicates that a system is supercritical and, if not changed, will increase in power. It must be noted the k_{eff} of the modeled reactor is seldom **one** due to modeling error even though its corresponding realistic problem is critical. Compared to Eq. (2.1), the dimensionality of Eq. (2.6) is reduced by one. However, this 6-dimension equation is still hard enough to solve.

2.1.3 Feedback in the Neutron Transport Equation

The NTE becomes considerably more difficult to solve because of the inherent multiphysics in a reactor. The most important forms of feedback come from such physics as TH and nuclide transmutation. If the feedback from other physics is taken into account, the macroscopic cross sections must be written with a new dependent variable as

$$\Sigma_i(\boldsymbol{r},E,t)\to\Sigma_i(\boldsymbol{r},\boldsymbol{S},E,t)=\sum_e N_e(\boldsymbol{S},E,t)\sigma_i^e(\boldsymbol{r},\boldsymbol{S},E,t)\quad i=t,s,f^{\cdot}, \tag{2.7}$$

where $\boldsymbol{S}$ is the vector of state variables determined by other equations that cause a change in the macroscopic cross section. For example the water density will affect the atomic number density, N_e, and the temperature can determine the microscopic cross section, σ_e, where e is the index of the elements.

The equations to determine the state variables will typically have some terms that depend on the solution of the NTE, hence we refer to this process as feedback. Examples of quantities in $\boldsymbol{S}$ can be temperature, moderator density, equilibrium Xenon, boron concentration, etc. The state variables are obtained from the solutions of other physics equations. The coupled equations are expressed in operator form as:

$$\mathcal{H}\left(\boldsymbol{r}, P(S_F), \phi, \Sigma_i, \boldsymbol{S}, k_{eff}, t\right) \boldsymbol{S}(\boldsymbol{r}) = 0\,, \tag{2.8}$$

where $\mathcal{H}$ is the operator dependent on the state variables, transport solutions and time. Variable t is present to generalize the time-dependent perturbation and the differential operator in time. k_{eff} is included to show the equation for the physics problem such as critical boron search. $P(S_F)$ is the thermal power generated by the fission.

For example, a simplified model to determine the temperature distribution is written as:

$$\rho c_p \frac{\partial T(t, \boldsymbol{r})}{\partial t} = \nabla \cdot k(\boldsymbol{r}, T)\nabla T + q'''(\boldsymbol{r}, S_F)\,, \tag{2.9}$$

where q''' is the heat generation rate. And the equations for the concentrations of the Iodine-135 and Xenon-135 are

$$\frac{dN_I}{dt} = \gamma_I S_F - \lambda_I N_I\,, \tag{2.10a}$$

$$\frac{dN_{Xe}}{dt} = \lambda_I N_I + \gamma_{Xe} S_F - \lambda_{Xe} N_{Xe} - \int_0^\infty \sigma_{X_e,a}(E') N_{Xe} \phi(\boldsymbol{r}, E') dE'\,. \tag{2.10b}$$

γ_I and γ_{Xe} are the effective fractions of fission products for I and Xe, respectively. And λ is the decay constant. The full sets of the equations of other physics are omitted, since they are out of scope in this thesis.

2.2 Overview of the Principle Methods

The realistic neutron transport problem can only be solved with the help of numerical methods. Two different classes of numerical methods have been developed in the field–the stochastic method and the deterministic method.

2.2.1 Monte Carlo Methods

The stochastic, or Monte Carlo (MC), methods simulate the behavior of individual particles representing the neutrons in the nuclear reactor [9, 58, 10, 59]. During the iteration, the MC code randomly samples the probability distributions of all aspects of the phase-space for a given problem, i.e., the birth of a fission neutron, the direction of flight, the energy of the neutron, the distance to the next collision and the type of reactions, etc. The MC code tracks the behavior of a neutron till the neutron leaks out of the reactor or is absorbed. If the absorption causes a fission, the fission sites will be recorded and used for the next iteration. The process is repeated for all the neutrons in the same iteration, that is referred to as batch. The iterative k-eigenvalue can be calculated by ratio, that is the number of new-generated neutrons simulated for the next batch to the number of neutrons simulated for the current batch.

Since the number of particles simulated is far smaller than the number of neutrons in the nuclear reactor and a finite number of batches are simulated, the results provided by the MC method are the estimates of the solutions associated with statistical uncertainties. Since much fewer assumptions and approximations are made in the MC codes compared to the deterministic codes, the MC solutions are widely considered as being able to provide the most accurate results. However, for the large-scale problem, the number of neutron particles simulated is tremendously large in order to acquire the results with reasonable uncertainties. Meanwhile, the computational time scales with the amount of the tallied quantities. MC methods also have the added challenge that the randomness of the events makes locating the position of the data in memory more time-consuming because modern memory architectures assume high degrees of spatial and temporal coherency in the memory accesses of programs. Therefore, Monte Carlo codes are still considered to be too expensive for the whole-core multiphysics transport calculation.

2.2.2 Deterministic Methods

The deterministic codes make many more approximations to solve the neutron transport equation compared with MC methods. The accuracy and efficiency of the codes are highly affected by the assumptions made in the numerical methods. The methods based on the Method of Characteristics (MOC) have been the most popular in recent decades [60, 7]. Due to high computational requirements of the direct 3D-MOC calculation, a variant of MOC, the 2D/1D method [1, 2, 3, 4, 41] has been popular in recent years.

The 2D/1D method takes advantage of the fact that the most significant heterogeneity in a Light Water Reactor (LWR) is in the radial direction, and there is less heterogeneity in the axial direction of a typical reactor core design. The essential idea of the 2D/1D method is to perform a highly accurate transport solution radially and a lower-fidelity nodal transport or diffusion solution axially. The 2D MOC is usually used as the radial solver, and axial solvers include various 1D methods, such as the Nodal Expansion Method (NEM), SP_N, S_N, etc [61]. The coupling between the 2D and 1D solvers involves the radial and axial transverse leakage terms [61, 62]. The Michigan Parallel

Characteristics Transport (MPACT) code has the capability of modeling with either a 3D-MOC calculation, or a 2D/1D calculation, therefore it is the representative of the modern deterministic high-fidelity codes [61, 63].

The methods presented in this thesis are implemented in MPACT. Therefore, the focus of the background is on the deterministic methods used in MPACT. Many of the details of the 3D-MOC scheme and 2D/1D scheme can be found in [1, 2, 4, 41, 61, 63, 64, 40, 65], so they are omitted here. However, the salient traits relative to this dissertation are described in the following sections.

2.2.3 Common Approximations for Deterministic Codes

2.2.3.1 Multigroup Approximation

Multigroup approximation is one of the most common approaches to discretize the energy space. In the multigroup approximation, the energy space is separated into G groups. The energy group is indexed with the conventional notation g ranging from 0 to G.

The bounds of the g^{th} energy group are $[E_g, E_{g-1}]$, with $E_0 = E_{max}$ and $E_G = 0$, where E_{max} is the maximum energy considered for a neutron, typically, 20 MeV. In the energy space, the scalar and angular fluxes are discretized into G groups, and the g^{th} component is defined as:

$$\varphi_g(\boldsymbol{r}, \boldsymbol{\Omega}) = \int_{E_g}^{E_{g-1}} \varphi(\boldsymbol{r}, \boldsymbol{\Omega}, E) dE \,, \tag{2.11}$$

$$\phi_g(\boldsymbol{r}) = \int_{E_g}^{E_{g-1}} \phi(\boldsymbol{r}, E) dE \,. \tag{2.12}$$

The multigroup cross section is defined by preserving the reaction rate:

$$\Sigma_{i,g}(\boldsymbol{r}) \approx \frac{\int_{E_g}^{E_{g-1}} \Sigma_{i,g}(\boldsymbol{r}, E) w(\boldsymbol{r}, E) dE}{\int_{E_g}^{E_{g-1}} w(\boldsymbol{r}, E) dE} \,, \; i = t, s, g, \ldots \,. \tag{2.13}$$

A suitable weighting function, w, has been introduced. The w is typical the scalar flux solution of the self-shielding calculations [66]. Usually, $\nu\Sigma_f$ are provided together with group-dependence, i.e.

$$\nu\Sigma_{f,g}(\boldsymbol{r}) \approx \frac{\int_{E_g}^{E_{g-1}} \nu(\boldsymbol{r}, E) \Sigma_{f,g}(\boldsymbol{r}, E) w(\boldsymbol{r}, E) dE}{\int_{E_g}^{E_{g-1}} w(\boldsymbol{r}, E) dE} \,. \tag{2.14}$$

The group-wise fission emission spectrum is obtained by integrating the energy along the interval, written as:

$$\chi_g(\boldsymbol{r}) = \int_{E_g}^{E_{g-1}} \chi(\boldsymbol{r}, E) dE \,. \tag{2.15}$$

And for the neutron velocity

$$v_g = \frac{\int_{E_g}^{E_{g-1}} dE}{\int_{E_g}^{E_{g-1}} \frac{1}{v(E)} dE} \,. \tag{2.16}$$

Next the time-dependent multigroup neutron transport equation is obtained by integrating Eq. (2.1) over each group's domain, and is written as:

$$\begin{aligned} \frac{1}{v_g}\frac{\partial \varphi_g\left(\boldsymbol{r},\boldsymbol{\Omega},t\right)}{\partial t} &+ \Sigma_{t,g}(\boldsymbol{r},t)\varphi_g\left(\boldsymbol{r},\boldsymbol{\Omega},t\right) + \boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\varphi_g\left(\boldsymbol{r},\boldsymbol{\Omega},t\right) \\ &= \sum_{g'=1}^{G}\int_0^{4\pi}\Sigma_{s,g'\to g}\left(\boldsymbol{r},\boldsymbol{\Omega}\cdot\boldsymbol{\Omega}',t\right)\varphi_{g'}(\boldsymbol{r},\boldsymbol{\Omega}',t)d\boldsymbol{\Omega}' \\ &+ \frac{1}{4\pi}\chi_{p,g}(\boldsymbol{r},t)\Big[1-\beta\left(\boldsymbol{r},t\right)\Big]S_F\left(\boldsymbol{r},t\right) + \frac{\chi_{d,g}(\boldsymbol{r},t)S_d\left(\boldsymbol{r},t\right)}{4\pi} \,, \end{aligned} \tag{2.17}$$

with the total fission source rewritten as

$$S_F(\boldsymbol{r},t) = \frac{1}{k_{eff}^0}\sum_{g=1}^{G}\nu\Sigma_{f,g}(\boldsymbol{r})\phi_g\left(\boldsymbol{r}\right) \,. \tag{2.18}$$

Following the same procedure, the steady-state multigroup neutron transport equation is

$$\begin{aligned} \boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\varphi_g\left(\boldsymbol{r},\boldsymbol{\Omega}\right) + \Sigma_{t,g}(\boldsymbol{r})\varphi_g\left(\boldsymbol{r},\boldsymbol{\Omega}\right) = \frac{\chi_g(\boldsymbol{r})}{4\pi k_{eff}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}\left(\boldsymbol{r}\right)\phi_{g'}\left(\boldsymbol{r}\right) + \\ \sum_{g'=1}^{G}\int_0^{4\pi}\Sigma_{s,g'\to g}\left(\boldsymbol{r},\boldsymbol{\Omega}\cdot\boldsymbol{\Omega}'\right)\varphi_{g'}(\boldsymbol{r},\boldsymbol{\Omega}')d\boldsymbol{\Omega}' \,. \end{aligned} \tag{2.19}$$

2.2.3.2 TCP$_0$ Scattering

The neutron scattering microscopic cross section is a complicated function of the $\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega}$, and is expanded via Legendre polynomials [67]:

$$\sigma_s^i(\boldsymbol{r},\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega}) = \sum_{n=0}^{\infty}\frac{2n+1}{4\pi}P_n(\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega})\sigma_{sn,g'\to g}(\boldsymbol{r}) \,. \tag{2.20}$$

Here, i is the index of isotopes, and P_n is the n^{th} Legendre polynomial.

$$\sigma_{sn,g'\to g}^i(\boldsymbol{r}) \equiv \int_{4\pi}P_n(\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega})\sigma_{s,g'\to g}^i(\boldsymbol{r},\boldsymbol{\Omega}'\cdot\boldsymbol{\Omega})d\boldsymbol{\Omega}' \,. \tag{2.21}$$

Using high-order scattering moment significantly increases the computational cost. Therefore, in MPACT, it is common to use an isotropic scattering cross section, which means that all but the 0^{th}

scattering moment are neglected. Then the macroscopic cross sections used are

$$\Sigma_{s0,g'\to g} = \sum_i N_i \sigma^i_{s0,g'\to g}\,, \tag{2.22}$$

where N is the atomic number density.

The resulting steady-state transport equation becomes much easier to solve and is given by

$$\begin{aligned}&\Big[\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}+\Sigma_{t,g}(\boldsymbol{r})\Big]\varphi_g(\boldsymbol{r},\boldsymbol{\Omega}) = \\ &\qquad \frac{1}{4\pi}\sum_{g'=1}^{G}\Sigma_{s0,g'\to g}(\boldsymbol{r})\phi_{g'}(\boldsymbol{r}) + \frac{\chi_g(\boldsymbol{r})}{4\pi k_{eff}}\sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\boldsymbol{r})\phi_{g'}(\boldsymbol{r})\,,\end{aligned} \tag{2.23}$$

and the time-dependent neutron transport equation is given by

$$\begin{aligned}&\frac{1}{v_g}\frac{\partial\varphi_g\left(\boldsymbol{r},\boldsymbol{\Omega},t\right)}{\partial t} + \Big[\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}+\Sigma_{t,g}(\boldsymbol{r})\Big]\varphi_g(\boldsymbol{r},\boldsymbol{\Omega}) = \\ &\frac{1}{4\pi}\sum_{g'=1}^{G}\Sigma_{s0,g'\to g}(\boldsymbol{r})\phi_{g'}(\boldsymbol{r}) + \frac{\chi_{p,g}(\boldsymbol{r},t)\Big[1-\beta\left(\boldsymbol{r},t\right)\Big]S_F\left(\boldsymbol{r},t\right)}{4\pi} + \frac{\chi_{d,g}(\boldsymbol{r},t)S_d\left(\boldsymbol{r},t\right)}{4\pi}\,.\end{aligned} \tag{2.24}$$

The isotropic approximation, however, makes the results less accurate. Therefore, Transport-Corrected P_0 (TCP$_0$) method is used [68, 69]. The scattering cross section matrix and the total cross section are artificially altered to offset losses in accuracy. The approach may result in negative cross sections and cause convergence problems for the transport solver.

Since all the numerical simulations in this thesis are performed with TCP$_0$, the TCP$_0$ correction is applied to the scattering term for all the derivations shown later in this chapter.

2.2.3.3 Angular Discretization

In the angular space, the discrete ordinates method or S_N [70], is used to discretize the angular variable. The angular space is discretized into P representative directions of travel. Each direction $\boldsymbol{\Omega}_p$ has a corresponding weight w_p that can be used for numerical integration over the angular variable as follows:

$$\int_{4\pi} f(\boldsymbol{\Omega})d\boldsymbol{\Omega} \approx \sum_{p=1}^{P} w_p f(\boldsymbol{\Omega}_p)\,. \tag{2.25}$$

Then the scalar flux is given by

$$\phi_g(\boldsymbol{r}) = \sum_{p=1}^{P} w_p \varphi_{g,p}(\boldsymbol{r})\,. \tag{2.26}$$

The S_N form of neutron transport equation can now be written as

$$\Big[\boldsymbol{\Omega}_p \cdot \boldsymbol{\nabla} + \Sigma_{t,g}(\boldsymbol{r})\Big]\varphi_{g,p}(\boldsymbol{r}) = \frac{1}{4\pi} q_g(\boldsymbol{r})\,, \tag{2.27}$$

where, $q_g(\boldsymbol{r})$ is the source and is defined by:

$$q_g(\boldsymbol{r}) \equiv \sum_{g'=1}^{G} \Sigma_{s0,g'\to g}(\boldsymbol{r})\phi_{g'}(\boldsymbol{r}) + \frac{\chi_g(\boldsymbol{r})}{4\pi k_{eff}} \sum_{g'=1}^{G} \nu\Sigma_{f,g'}(\boldsymbol{r})\phi_{g'}(\boldsymbol{r}) + Q_{ext,g}(\boldsymbol{r})\,, \tag{2.28}$$

where $Q_{ext,g}$ is the source from time discretization, delayed neutron precursor, etc.

Eq. (2.27) is further solved with the MOC method. Along the direction $\boldsymbol{\Omega}_p$, Eq. (2.27) is written as

$$\Big[\frac{d}{ds} + \Sigma_{t,g}(\boldsymbol{r}_{p,0} + s\boldsymbol{\Omega}_p)\Big]\varphi_{g,p}(\boldsymbol{r}_{p,0} + s\boldsymbol{\Omega}_p) = \frac{1}{4\pi} q_g(\boldsymbol{r}_{p,0} + s\boldsymbol{\Omega}_p)\,, \tag{2.29}$$

that is formulated as an ordinary differential equation. Here $\boldsymbol{r}_{p,0}$ is an arbitrary reference point.

The equation Eq. (2.29) can be solved exactly provided the exact source q_g and cross sections. In practice, the problem is discretized into a lot of fine spatial meshes, and cross sections Σ and the source q_g are assumed to be constants for each spatial region. Recently, a linear source approximation has been introduced to capture the spatial variation of the source inside each cell and provide better accuracy and efficiency [71, 72, 73]. The flat source approximation or linear source approximation is valid when the mesh size is small. The corresponding mesh is referred to as the *fine mesh* in this thesis.

It should be noted, in real problems, q is not predefined. It is obtained via iteration.

2.3 Steady-State Iteration Methods

2.3.1 Source Iteration

Source iteration is the simplest method and the most basic method to solve the NTE. And either 2D/1D method or the 3D-MOC method can be generalized as a source iteration process. For simplicity, the steady-state k-eigenvalue neutron transport equation is represented in operator notation with λ-eigenvalue as:

$$\mathcal{L}\boldsymbol{\varphi} = \frac{1}{4\pi}\Big[\mathcal{S} + \lambda\boldsymbol{\chi}\mathcal{F}\Big]\boldsymbol{\phi}\,, \tag{2.30}$$

with

$$\boldsymbol{\phi} = \mathcal{M}_0\boldsymbol{\varphi}\,, \tag{2.31a}$$

$$\lambda \equiv \frac{1}{k_{eff}}\,. \tag{2.31b}$$

Here

$$\Big[\mathcal{L}\Big]_g = \Big[\boldsymbol{\Omega}\cdot\boldsymbol{\nabla} + \Sigma_{t,g}(\boldsymbol{r})\Big]\,, \tag{2.31c}$$

$$\Big[\mathcal{S}\Big]_g = \sum_{g'=1}^{G}\Sigma_{s0,g'\to g}(\boldsymbol{r})(\cdot), \tag{2.31d}$$

$$\mathcal{F} = \sum_{g'=1}^{G}\nu\Sigma_{f,g'}(\boldsymbol{r})(\cdot)\,, \tag{2.31e}$$

$$\Big[\boldsymbol{\chi}\Big]_g = \boldsymbol{\chi_g}(\boldsymbol{r})(\cdot)\,, \tag{2.31f}$$

$$\mathcal{M}_i = \int_{4\pi}\boldsymbol{\Omega}^i(\cdot)d\boldsymbol{\Omega} \approx \sum_{p=1}^{G}\omega_p\boldsymbol{\Omega}_p^i(\cdot), i = 1,2\cdots\,. \tag{2.31g}$$

$\boldsymbol{\phi}$ and $\boldsymbol{\varphi}$ are the vectors of the group-wise scalar fluxes and angular fluxes, respectively. The basic approach to solve the neutron transport problem is the source iteration scheme illustrated in Algorithm 1. To avoid misleading, we use the script $^{(n)}$ to index the iteration, and ^{n} for the n^{th} power. Though such details of the source iteration as the procedure of MOC and 2D/1D scheme are omitted, the algorithm can still show the overall procedure of source iteration without loss of generality. The source iteration is equivalent to the power iteration method for the solution of k-eigenvalue problems.

Generally, the convergence rate of the source iteration scheme is prohibitively slow for practical problems, and should never be the method used for realistic problems. The modern iterative methods make use of convergence acceleration schemes [74, 39]. However, in the solution of the

Algorithm 1 Source iteration scheme for k-eigenvalue neutron transport problem.

```
Input maximum outer iteration number $N$, inner iteration number $N_1$; tolerance $\epsilon_c$, $\epsilon_f$; initial guess $\boldsymbol{\phi}^{(0)}$, $\lambda^{(0)}$.
for $n = 0, 1, \ldots, N$ do
    Let $\boldsymbol{\phi}^{(n,0)} \leftarrow \boldsymbol{\phi}^{(n)}$, and calculate the fission source
        $\mathcal{Q}^{(n)} = \lambda^{(n)} \boldsymbol{\chi} \mathcal{F} \boldsymbol{\phi}^{(n)}\,.$    (Alg 1.1)
    for $n_1 = 0, 1, \ldots, N_1$ do
        Compute the angular flux via transport sweep:
            $\mathcal{L} \boldsymbol{\varphi}^{(n,n_1+1)} = \frac{1}{4\pi} \left[ \mathcal{S} \boldsymbol{\phi}^{(n,n_1)} + \mathcal{Q}^{(n)} \right].$    (Alg 1.2)
        Update the Scalar flux
            $\boldsymbol{\phi}^{(n,n_1+1)} = \mathcal{M}_0 \boldsymbol{\varphi}^{(n,n_1+1)}\,.$    (Alg 1.3)
    end for
    Let $\boldsymbol{\phi}^{(n+1)} \leftarrow \boldsymbol{\phi}^{(n,N_1)}$.
    Update the eigenvalue
        $\lambda^{(n+1)} = \lambda^{(n)} \frac{\left\| \mathcal{F} \boldsymbol{\phi}^{(n)} \right\|}{\left\| \mathcal{F} \boldsymbol{\phi}^{(n+1)} \right\|}\,.$    (Alg 1.4)
    Let $k_{eff}^{(n+1)} \leftarrow 1/\lambda^{(n+1)}$.
    if $\left| k_{eff}^{(n+1)} - k_{eff}^{(n)} \right| < \epsilon_c$ and $\left\| \mathcal{F} \boldsymbol{\phi}^{(n)} - \mathcal{F} \boldsymbol{\phi}^{(n+1)} \right\| < \epsilon_f \left\| \mathcal{F} \boldsymbol{\phi}^{(n+1)} \right\|$ then
        break
    end if
end for
```

acceleration equations, the power iteration is still used. The next section describes the state-of-the-art acceleration method that is a major focus of this thesis.

2.3.2 CMFD Acceleration

One of the most popular acceleration methods for the NTE is the CMFD method. It is a variant of NDA for the general problems [13]. There are two distinct equations involved in the CMFD acceleration. The first is a high-order problem, that is the multigroup transport problem. The second is a low-order multigroup diffusion problem, that is solved with a much coarser spatial mesh. The spatial mesh where the solution of low-order diffusion problem is performed is referred to as the *coarse mesh*.

2.3.2.1 CMFD Equation for Steady-State Calculation

In the CMFD scheme, the low-order multigroup diffusion problem has the form:

$$\begin{aligned}-\nabla\cdot\mathbf{D}_g\nabla\phi_g(\boldsymbol{r})+\Sigma_{t,g}(\boldsymbol{r})\phi_g(\boldsymbol{r})-\sum_{g'=1}^{G}\Sigma_{s0,g'\to g}(\boldsymbol{r})\phi_{g'}(\boldsymbol{r})&\\=\lambda\frac{\chi_g(\boldsymbol{r})}{4\pi}\sum_{g'=1}^{G}\Sigma_{f,g'}(\boldsymbol{r})\phi_{g'}(\boldsymbol{r})\,.&\end{aligned}\tag{2.32}$$

The equation is obtained by integrating Eq. (2.23) over the whole angular space. Eq. (2.32) is also generalized so that its solution, $\Phi_g(r)$ is equivalent to that obtained from Eq. (2.31c).

In the CMFD problem, the domain is decomposed into M coarse cells, or coarse meshes, indexed with m. The solution of the low-order problem is coarse-cell-wise, while the high-order transport sweeps are performed on a much finer mesh. The derivation of the low-order diffusion problem for CMFD acceleration is illustrated as below.

Integrating Eq. (2.32) over the space of each coarse cell, the k-eigenvalue NTE is transformed into the neutron balance equation:

$$\begin{aligned}\sum_{m'\in N(m)}\Gamma_{(m,m')}J_{(m,m'),g}+\left(\Sigma_{t,m,g}\Phi_{m,g}-\sum_{g'=1}^{G}\Sigma_{s0,m,g'\to g}\Phi_{m,g'}\right)V_m&\\=\lambda\chi_g\sum_{g'=1}^{G}\Sigma_{f,m,g'}\Phi_{m,g'}V_m,\qquad 1\leq g\leq G,\quad 1\leq m\leq M\,.&\end{aligned}\tag{2.33}$$

Here (m,m') denotes the surface of the m^{th} coarse mesh and its neighboring m'^{th} cell with the outward-pointing normal vector to m'^{th} cell. $\Gamma_{(m,m')}$ is the area of the surface and V_m is the volume of the coarse cell. $N(m)$ represents the collection of the neighboring cells of mesh m. Φ is used to represent the coarse-mesh flux solutions.

The net current $J_{(m,m'),g}$ represents the net outflow of neutrons from coarse mesh m to m', and

$$J_{(m,m'),g} = -J_{(m',m),g} \,. \tag{2.34}$$

To obtain Eq. (2.33) as a diffusion equation, the traditional CMFD representation of the net current is introduced. Here the net current is calculated by the finite difference approximation of the generalized Fick's Law containing the nonlinear correction, $\hat{D}_{(m,m'),g}$:

$$J_{(m,m'),g} = -\tilde{D}_{(m,m'),g}(\Phi_{m',g} - \Phi_{m,g}) + \hat{D}_{(m,m'),g}(\Phi_{m,g} + \Phi_{m',g}) \,. \tag{2.35}$$

The diffusion coefficient on a surface is defined as:

$$\tilde{D}_{(m,m'),g} = \frac{2D_{m,g}D_{m',g}}{(D_{m,g} + D_{m',g})h_{m,m'}} \,, \tag{2.36}$$

and the nonlinear correction term is defined as

$$\hat{D}_{(m,m'),g} = \frac{J^{ts}_{(m,m'),g} + \tilde{D}_{(m,m'),g}(\bar{\Phi}^{ts}_{m,g} - \bar{\Phi}^{ts}_{m',g})}{\bar{\Phi}^{ts}_{m',g} + \bar{\Phi}^{ts}_{m,g}} \,. \tag{2.37}$$

In this fashion,

$$\hat{D}_{(m,m'),g} = -\hat{D}_{(m',m),g} \,, \tag{2.38a}$$

$$\tilde{D}_{(m,m'),g} = \tilde{D}_{(m',m),g} \,. \tag{2.38b}$$

The nonlinear correction term, $\hat{D}$, is used to enforce the equivalence of the diffusion balance equation with the transport equation. The superscript ^{ts} denotes the quantities are from the transport solution.

Inserting Eqs. (2.35) to (2.37) into Eq. (2.33) yields the neutron balance equation:

$$\begin{aligned}\sum_{m' \in N(m)} \Gamma_{m,m'} \Big[-\tilde{D}_{(m,m'),g}(\Phi_{m',g} - \Phi_{m,g}) + \hat{D}_{(m,m'),g}(\Phi_{m,g} + \Phi_{m',g}) \Big] \\ + \Big(\Sigma_{t,m,g}\Phi_{m,g} - \sum_{g'=1}^{G} \Sigma_{s0,m,g' \to g}\Phi_{m,g'} \Big) V_m = \lambda \chi_g \sum_{g'=1}^{G} \Sigma_{f,m,g'}\Phi_{m,g'}V_m \,, \\ 1 \le g \le G, \quad 1 \le m \le M \,.\end{aligned} \tag{2.39}$$

Eq. (2.39) is the multigroup CMFD equation for the k-eigenvalue problem.

The coefficients of Eq. (2.37) and Eq. (2.39) are homogenized from the k-eigenvalue neutron transport equation in the flux volume-weighted way [32, 51]:

$$\bar{\Phi}^{ts}_{m,g} = \frac{\int_{\boldsymbol{r}\in C_m} \phi^{ts}_g(\boldsymbol{r})dV}{V_m} \tag{2.40a}$$

$$\Sigma_{x,m,g} = \frac{\int_{\boldsymbol{r}\in C_m} \Sigma_{x,g}(\boldsymbol{r})\phi^{ts}_g(\boldsymbol{r})dV}{\int_{\boldsymbol{r}\in C_m} \phi^{ts}_g(\boldsymbol{r})dV}, \;\; \Sigma_x = \Sigma_t, \Sigma_{s0}, \Sigma_f \ldots\,, \tag{2.40b}$$

$$\chi_{m,g} = \frac{\int_{\boldsymbol{r}\in C_m} \chi_g(\boldsymbol{r}) \sum_{g'=1}^{G} \nu\Sigma^{ts}_{f,g'}\phi^{ts}_{g'}(\boldsymbol{r})dV}{\int_{\boldsymbol{r}\in C_m} \sum_{g'=1}^{G} \nu\Sigma^{ts}_{f,g'}\phi^{ts}_{g'}(\boldsymbol{r})dV}\,, \tag{2.40c}$$

where C_m is the domain of the m^{th} coarse cell. Writing the solution as a vector, the 3D CMFD k-eigenvalue equation in operator notation is

$$(\mathbf{M} - \mathbf{S})\mathbf{\Phi} = \lambda\boldsymbol{\chi}\mathbf{F}\mathbf{\Phi}\,. \tag{2.41}$$

This is a generalized eigenvalue problem with $\mathbf{\Phi} \in \mathbb{R}^{MG}$, $\mathbf{M}$, $\mathbf{S} \in \mathbb{R}^{MG\times MG}$, $\mathbf{F} \in \mathbb{R}^{M\times MG}$ and $\boldsymbol{\chi} \in \mathbb{R}^{MG\times M}$.

When $\mathbf{\Phi}$ is ordered in node-major fashion, it is written as:

$$\mathbf{\Phi} = [\Phi_{1,1}\cdots\Phi_{1,G}\cdots\Phi_{M,g}\cdots\Phi_{M,G}]^T\,. \tag{2.42}$$

The operators are then defined in a similar way as

$$\begin{aligned}[\mathbf{M\Phi}]_{(m-1)G+g} &= \Sigma_{t,m,g}\Phi_{m,g} \\ &+ \sum_{m'\in N(m)} \frac{\Gamma_{m,m'}}{V_m}\Bigg[-\tilde{D}_{(m,m'),g}(\Phi_{m',g} - \Phi_{m,g}) + \hat{D}_{(m,m'),g}(\Phi_{m,g} + \Phi_{m',g})\Bigg]\,,\end{aligned} \tag{2.43a}$$

$$[\mathbf{S\Phi}]_{(m-1)G+g} = \sum_{g'=1}^{G} \Sigma_{s0,m,g'\to g}\Phi_{m,g'}\,, \tag{2.43b}$$

$$[\mathbf{F\Phi}]_m = \sum_{g'=1}^{G} v\Sigma_{f,m,g'}\Phi_{m,g'}\,, \tag{2.43c}$$

$$[\boldsymbol{\chi}\mathbf{F\Phi}]_{(m-1)G+g} = \chi_{m,g}\sum_{g'=1}^{G} v\Sigma_{f,m,g'}\Phi_{m,g'}\,. \tag{2.43d}$$

For boundary cells, typical Marshak boundary conditions are used. $\hat{D}$ and $\tilde{D}$ are then modified to take the boundary conditions into account.

Since Eq. (2.41) is an eigenvalue problem, a normalization condition must be applied to ensure a unique solution. The typical normalization is defined so that $||\mathbf{F\Phi}||_2 = C$, where C is a constant.

2.3.2.2 CMFD Acceleration

Once the low-order diffusion problem is solved, the low-order solution $\mathbf{\Phi}$ is used to update the fine-mesh scalar flux by:

$$\phi_g^{corr}(\mathbf{r}) = \frac{\Phi_{m,g}}{\bar{\Phi}_{m,g}^{ts}}\phi(\mathbf{r})\,,\ \mathbf{r} \in C_m\,. \tag{2.44}$$

The corrected fine mesh solution is used to calculate the scattering source and the fission source with the eigenvalue calculated in low-order calculation. The angular flux at the boundary is also scaled by using Eq. (2.44) (or something similar). The algorithm of the CMFD acceleration is illustrated in Algorithm 2.

For the k-eigenvalue problem, CMFD acceleration reduces the spectral radius of the iteration scheme from nearly 1 to around 0.4 for a typical reactor [16, 75, 76].

Algorithm 2 CMFD acceleration for k-eigenvalue problems

```
Input maximum outer iteration number $N$, inner source iteration number $N_1$; tolerance $\epsilon_c$, $\epsilon_f$;
  initial guess $\boldsymbol{\phi}^{(0)}$, $J^{(0)}$, $\lambda^{(0)}$.
for $n = 0, 1, \ldots, N$ do
    Compute $\hat{D}$ by Eq. (2.37) using the current estimate in the transport system.
    Solve the CMFD eigenvalue problem Eq. (2.41) to obtain the scalar flux on the CMFD
      spatial grid $\mathbf{\Phi}^{(n+1/2)}$ and the eigenvalue $\lambda^{(n+1/2)}$.
    Update the fine mesh flux $\boldsymbol{\phi}^{(n)}$ via Eq. (2.44) and use corrected flux and eigenvalue $\lambda^{(n+1/2)}$
      to generate the fission source.
    Perform $N_1$ iterations (i.e. sweeps) on the transport problem. During the sweeps, the fission
      source and eigenvalue are fixed. The scalar fluxes on the transport grid and the current at
      the interfaces of CMFD cells are stored.
    Let $k_{eff}^{(n+1)} \leftarrow 1/\lambda^{(n+1/2)}$, $\boldsymbol{\phi}^{(n+1)} \leftarrow \boldsymbol{\phi}^{(n,N_1)}$.
    if $\left|k_{eff}^{(n+1)} - k_{eff}^{(n)}\right| < \epsilon_c$ and $\left\|\mathcal{F}\boldsymbol{\phi}^{(n)} - \mathcal{F}\boldsymbol{\phi}^{(n+1)}\right\| < \epsilon_f \left\|\mathcal{F}\boldsymbol{\phi}^{(n+1)}\right\|$ then
        break
    end if
end for
```

2.3.3 Methods for the Multigroup Diffusion Eigenvalue Problem

The problem of interest for the CMFD equation is the multigroup diffusion eigenvalue problem. In reactor physics, it is a generalized eigenvalue problem of the following form:

$$\mathbf{M}\boldsymbol{\phi} = \lambda\mathbf{F}\boldsymbol{\phi}\,. \tag{2.45}$$

In this general eigenvalue problem, only $\mathbf{M}$ is invertible. Note that $\mathbf{M}$ and $\mathbf{F}$ in Eq. (2.45) are the $\mathbf{M}-\mathbf{S}$ and $\boldsymbol{\chi}\mathbf{F}$ in Eq. (2.41), respectively. The goal of an eigenvalue problem is to compute the eigenvalue(s) λ and obtain the corresponding eigenvectors.

2.3.3.1 Wielandt Shifted Power Iteration

The simplest method to obtain the solution to Eq. (2.45) is Power Iteration (PI), which is guaranteed to converge to the smallest eigenvalue of the system. The detail is shown in Algorithm 3. The convergence rate of the PI is determined by the dominance ratio of the system:

$$\mathrm{DR}_{PI} = \frac{|\lambda_1|}{|\lambda_2|}\,. \tag{2.46}$$

Here, λ_1 is the smallest eigenvalue in magnitude, and λ_2 is the second smallest. For most problems in reactor physics, the dominance ratio of the system is close to 1, and PI converges too slowly for practical use.

Algorithm 3 Power iteration algorithm.

```
1: Input maximum power iteration number $L$, initial guess $\mathbf{\Phi}^{(0)}$, normalization factor $\Phi_0$ for
   solving Eq. (2.45).
2: for $l = 0, 1, 2, \cdots, L-1$ do
3:     Solve the WS linear system:

       $\mathbf{M}\mathbf{\Phi}^{\left(l+\frac{1}{2}\right)} = \lambda^{(l)}\mathbf{F}\mathbf{\Phi}^{(l)}\,,$    (Alg 3.1a)
       $\lambda^{(l+1)} = \lambda^{(l)}\frac{\left\|\mathbf{F}\mathbf{\Phi}^{(l)}\right\|}{\left\|\mathbf{F}\mathbf{\Phi}^{\left(l+\frac{1}{2}\right)}\right\|}\,.$    (Alg 3.1b)

4:     Normalize the eigenvector solution by

       $\mathbf{\Phi}^{(l+1)} = \frac{\mathbf{\Phi}^{\left(l+\frac{1}{2}\right)}}{\left\|\mathbf{\Phi}^{\left(l+\frac{1}{2}\right)}\right\|}\Phi_0\,.$    (Alg 3.1c)

5: end for
6: return $\lambda^{(L)}, \Phi^{(L)}$
```

Therefore, the more popular method is the Wielandt shifted (WS) Power iteration.

$$\mathbf{M} - \lambda_{\mathrm{s}}\mathbf{F}\boldsymbol{\phi} = (\lambda - \lambda_s)\mathbf{F}\boldsymbol{\phi}\,. \tag{2.47}$$

The algorithm is shown in Algorithm 4. Assume $\lambda_2 > 0$. The method can significantly reduce the dominance ratio of a system when the provided shift parameter λ_s is close to the eigenvalue of interest, since the dominance ratio becomes

$$\text{DR}_{WSPI} = \frac{|\lambda_1 - \lambda_s|}{|\lambda_2 - \lambda_s|} \,. \tag{2.48}$$

Algorithm 4 Wielandt shifted power iteration algorithm.

```
1: Input maximum power iteration number $L$, eigenvalue $\lambda^{(0)}$, initial guess $\mathbf{\Phi}^{(0)}$, normalization factor $\Phi_0$ for solving Eq. (2.45).
2: for $l = 0, 1, 2, \cdots, L-1$ do
3:     Determine the WS parameter $\lambda_s^{(l)}$.
4:     Solve the WS linear system:
           $\left[\mathbf{M} - \lambda_s^{(l)}\mathbf{F}\right]\mathbf{\Phi}^{\left(l+\frac{1}{2}\right)} = \left[\lambda^{(l)} - \lambda_s^{(l)}\right]\mathbf{F}\mathbf{\Phi}^{(l)} \,,$   (Alg 4.1a)
           $\lambda^{(l+1)} = \lambda_s^{(l)} + \left[\lambda^{(l)} - \lambda_s^{(l)}\right]\frac{\left\|\mathbf{F}\mathbf{\Phi}^{(l)}\right\|}{\left\|\mathbf{F}\mathbf{\Phi}^{\left(l+\frac{1}{2}\right)}\right\|} \,.$   (Alg 4.1b)
5:     Normalize the eigenvector solution by
           $\mathbf{\Phi}^{(l+1)} = \frac{\mathbf{\Phi}^{\left(l+\frac{1}{2}\right)}}{\left\|\mathbf{\Phi}^{\left(l+\frac{1}{2}\right)}\right\|}\Phi_0 \,.$   (Alg 4.1c)
6: end for
7: return $\lambda^{(L)}, \Phi^{(L)}$
```

In order to achieve a minimum spectral radius in Algorithm 4, one must choose λ_s as close to λ as possible. However, for practical simulation, λ_s cannot be very close. Otherwise, over-shift ($\lambda_s \geq \lambda$) may happen, and the converged λ is not the smallest eigenvalue. Or the condition number of the linear system $\mathbf{M} - \lambda_s\mathbf{F}$ becomes very large, and the linear system becomes hard to solve for iterative solvers [77].

In practice, two approaches are taken: (1) choosing some reasonable fixed value for λ_s, and (2) defining $\lambda_s^{(l)}$ as a function of the current eigenvalue estimate $\lambda^{(l)}$.

A thorough review of the methods to determine λ_s has been provided, and more sophisticated schemes have been proposed in[78]. In MPACT, we typically choose $\lambda_{min} = 1/3$. The Purdue Advanced Reactor Core Simulator (PARCS) shift [6] and the more advanced methods from [78] are also available.

2.3.3.2 Multilevel-in-Space-and-Energy Diffusion (MSED) Method

There are other eigenvalue solvers such as multilevel diffusion solvers [32, 79, 80]. These methods reduce the computational cost of solving the eigenvalue by preconditioning the system. The multilevel solvers treat the multigroup diffusion problem as a high-order problem, and construct additional low-order problems on coarser spatial grids, coarser energy grids or both.

Algorithm 5 Simplified algorithm of MSED solver.

```
Input $L_{MG}$, $L_{GR}$
for $l_{mg} = 1, 2, \cdots L_{MG}$ do
    Collapse the operators $\mathbf{M}_{MG},\ \mathbf{F}_{MG}$ in multigroup CMFD equations to the operators
    $\mathbf{M}_{1G},\ \mathbf{F}_{1G}$ and obtain initial guess of the solution $\mathbf{\Phi}_{1G}$ in grey diffusion equations using
    the latest multigroup fluxes $\mathbf{\Phi}_{MG}$
    Solve the grey diffusion eigenvalue equation:
        $\mathbf{M}_{1G}\mathbf{\Phi}_{1G} = \lambda_{1G}\mathbf{F}_{1G}\mathbf{\Phi}_{1G}\,,$    (Alg 5.1)
    with Wielandt shift parameter $\lambda_s$, and $L_{GR}$ grey power iterations.
    Solve the multigroup fixed source problem
        $\mathbf{M}_{MG}\mathbf{\Phi}_{MG} = \lambda_{1G}\boldsymbol{\chi}_{MG}\mathbf{F}_{1G}\mathbf{\Phi}_{1G}\,,$    (Alg 5.2)
    where $\boldsymbol{\chi}_{MG}$ is the matrix form of the multigroup fission spectrum.
    Update the eigenvalue $\lambda$.
    Check the convergence of the residual of CMFD Solutions.
    if converged then
        break
    end if
end for
return $\lambda, \mathbf{\Phi}_{MG}$
```

The multilevel solver of interest in this work is the MSED solver. It is a multilevel method composed of a fixed source multigroup diffusion equation and a grey diffusion eigenvalue problem. The Wielandt shifted power iterations are performed for the grey diffusion equation, where the coefficients are collapsed from the multigroup diffusion equation. For MSED, the number of energy groups of the grey diffusion equation is *one*, in other works the equation is typically two-groups. However, one-group is preferred here because the fission emission spectrum is only dependent on the energy integrated fission source. Therefore, the grey equation is a one-group CMFD (1GCMFD) equation, and the results are used to approximate the fission source of the multigroup diffusion equation. The multigroup fluxes are obtained from the multigroup fixed source CMFD problem. Additionally, in the spatial domain, the MSED algorithm utilizes the multigrid [81] method to solve the linear system of the diffusion problem. Several iterations between the multigroup level and the grey diffusion level are performed until the convergence criteria of the MSED solver is met. A simplified version of the algorithm for the MSED solver is shown in Algorithm 5. L_{MG} is the number of iterations performed for the MGCMFD system, and L_{GR} is the number of power iterations for the 1GCMFD system. Compared to the standard MGCMFD solver in MPACT, the computational cost of MSED is much less. Moreover, the maximum number of power iterations performed in MSED method is $L_{MG}L_{GR}$, that is much larger than the number of WS power iterations to solve the MGCMFD system in the default method. Therefore, MSED helps the iteration scheme converge faster.

2.3.4 Other Solvers

There are other solvers such as the Generalized Davidson method [82] and Jacobian-free Newton-Krylov (JFNK) method [83]. The performance of these methods in practical high-fidelity simulations remains to be investigated.

2.3.5 Coupling with Other Physics Solver

2.3.5.1 Picard Iteration

In the state-of-art nuclear reactor modeling, the feedback from other physics is taken into account. The neutronics problem is usually coupled with the feedback (e.g. Thermal Hydraulics) via a Picard iteration [84, 26, 3, 23, 4, 24, 25]. The Picard iteration is the preferred multiphysics iteration in these tools because frequently the physics codes are developed separately, and need to function independently, of each other. Consequently, the major whole-core, high-fidelity simulation codes: DeCART [2], nTRACER [3], STREAM [41], NECP-X [33] and MPACT/VERA-CS [4] use this approach. In these codes, the multiphysics coupling occurs at the "outer" iteration in which a CMFD-accelerated transport iteration is performed, then the thermal-hydraulics (TH) solver performs the solution. The appropriate solution fields are exchanged between the physics after each iteration. The flowchart of the implementation of the Picard iteration scheme is shown in Figure 2.1. It should be noted that the Picard scheme presented here is slightly different from what has been presented in [20, 85], where the neutronics solver is fully converged and then coupled with solvers of other physics.

In realistic simulations, it has been observed that this iteration scheme is not stable. As a result, the under-relaxation has to be used. The under-relaxation may be applied to any number of state variables. The most common approach is to apply relaxation to the power distribution. Other common forms of relaxation may also be applied to the material density or material temperatures. If n is used as the index of the outer iteration, then the pin power distribution $p_{tr}^{(n)}(\mathbf{r})$ that is transferred

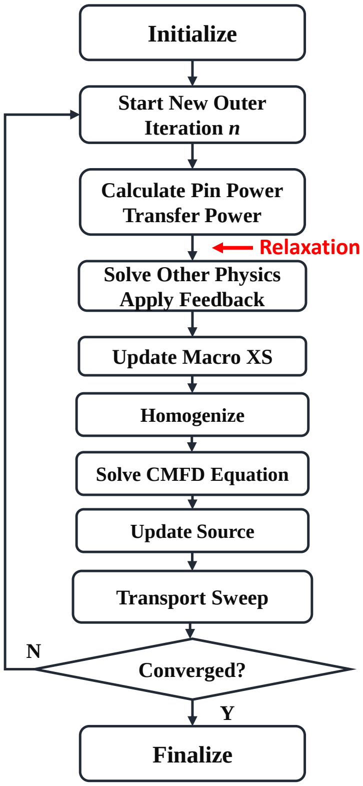


Figure 2.1: Flowchart of multiphysics scheme for steady-state calculation in MPACT. This scheme is also used in codes DeCART, nTRACER, STREAM and NECP-X. "XS" stands for cross section.

to the other physics solver is calculated by

$$p_{tr}(\mathbf{r}) = (1-\beta)p^{n-1}(\mathbf{r}) + \beta p^{(n)}(\mathbf{r}) \, , \tag{2.49}$$

where β is the relaxation factor and $p^{(n)}(\mathbf{r})$ is the pin power distribution calculated by the neutronics solver at the n^{th} outer iteration. There is no standard way to determine the relaxation factor, therefore, the factor is invariably chosen in an empirical way. Sufficient and optimal relaxation factors are known to be problem-dependent. Thus, in practice, it is quite common to encounter problems that converge considerably slower or diverge compared to the non-feedback cases.

For this Picard iteration scheme, an important and perhaps counter-intuitive observation made recently with respect to the use of CMFD acceleration, is that when one more tightly converges the low-order diffusion solutions, the coupled iteration scheme converges more slowly and becomes less stable. As better acceleration methods for neutronics problems are being developed for problems without feedback, the tight convergence of the low-order diffusion solutions creates the issue of having a less robust multiphysics iteration scheme. This ultimately leads to a penalty on the convergence rate of the acceleration scheme [82, 32].

2.3.5.2 Other Iteration Methods

Besides the Picard iteration, other methods have been developed and introduced to coupling the neutronics and other feedback. These methods include the JFNK [36, 20] method and the Anderson acceleration (AA) [37]. The JFNK is a modified newton method without explicitly formulating the Jacobian matrix. The implementation of JFNK, however, still requires significant changes to the current iterations schemes and the computational efficiency for the neutron transport calculation is still limited [86]. The AA is a method that mixes the solutions of previous several outer iterations adaptively to stabilize the calculation. Compared to the JFNK method, it is much easier to implement. Early results in [87, 19] did not show a substantial benefit from AA. Results from [38] in nTRACER, have shown AA helps to achieve much better stability and efficiency and is potential for practical simulations in the future.

2.3.6 Stability of Iteration schemes

Stability is used in a lot of chapters in this thesis. In the steady-state problem, it is used to show whether the iterative method can converge. A stable method is a method that with the number of iterations taken increasing, the iterative solution $\phi^{(n)}(x)$ approaches and eventually converges to the actual solution of the discretized steady-state problem $\phi(x)$. Particularly, in a mathematics form, we want a method to be stable so that:

$$\lim_{n\to\infty} \phi^{(n)}(x) = \phi(x)\,. \tag{2.50}$$

The stability for the iterative scheme is different from the stability for the time-dependent solver that will be presented later in this chapter.

2.4 Transient Methods in MPACT

The Transient Multilevel (TML) method is used in MPACT to simulate the transient behavior of the reactor, and is briefly introduced here. More details can be found in [54]. The TML scheme is a multilevel algorithm using three levels with 3D-transport coupled to 3D-CMFD, and 3D-CMFD coupled to the EPKEs. This effectively captures the evolution of the flux. In each level, the flux equation is predicted with a coarse predictor time step, and corrected with a fine corrector time step. This requires the flux factorized as a shape function and amplitude function. The shape function is assumed to vary more slowly than the amplitude function in time. In the 3D-transport/3D-CMFD coupling, the angular flux is factored as a shape function in angular space multiplying a cell-wise amplitude function. The angular and sub-pin flux shape distribution are computed in transport calculation first with a coarse time step. Then the cell-wise amplitude function will be corrected by the cell-wise scalar flux calculated from the CMFD equations using multiple finer steps. In the 3D-CMFD/EPKEs level, the cell-wise amplitude function of each CMFD step is treated as a shape function, and the whole-core amplitude function is corrected by the EPKEs solution. The EPKEs are solved with finer time steps.

2.4.1 Multigroup Transient Fixed Source Problem

To start, the multigroup time-dependent NTE is written as:

$$\begin{aligned}&\frac{1}{v_g}\frac{\partial\varphi_g\left(\boldsymbol{r},\boldsymbol{\Omega},t\right)}{\partial t}+\Big[\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}+\Sigma_{t,g}(\boldsymbol{r})\Big]\varphi_g(\boldsymbol{r},\boldsymbol{\Omega})=\\&\frac{1}{4\pi}\sum_{g'=1}^{G}\Sigma_{s0,g'\to g}(\boldsymbol{r})\phi_{g'}(\boldsymbol{r})+\frac{\chi_{p,g}(\boldsymbol{r},t)\Big[1-\beta\left(\boldsymbol{r},t\right)\Big]S_F\left(\boldsymbol{r},t\right)}{4\pi}+\frac{\chi_{d,g}(\boldsymbol{r},t)S_d\left(\boldsymbol{r},t\right)}{4\pi}\,.\end{aligned} \tag{2.24 revisited}$$

The time domain must be discretized for numerical simulation. In this section, n is used to index the time point rather than the outer iteration. The outer iteration is indexed with ℓ instead.

$$x^{(n)}=x(t_n)\,, \tag{2.51}$$

i.e., $x^{(n)}$ is used to represent the variable x at the n^{th} time point t_n. The n^{th} time step size at the time t_n is Δt_n. The time derivative term in Eq. (2.24) is discretized with the BE method using the isotropic approximation

$$\frac{1}{v_g}\frac{\partial\varphi_g^{(n)}(\boldsymbol{r},\boldsymbol{\Omega})}{\partial t}\approx\frac{1}{4\pi v_g}\frac{\partial\phi_g^{(n)}(\boldsymbol{r})}{\partial t}\approx\frac{\phi_g^{(n)}(\boldsymbol{r})-\phi_g^{(n-1)}(\boldsymbol{r})}{4\pi v_g\Delta t_n}\,. \tag{2.52}$$

The isotropic approximation is used to overcome the practical difficulty for explicitly solving and storing the angularly dependent time-derivative term, and is reasonably accurate for practical simulations when Δt_n is small [44]. The exponential transformation method is also applied to treat the time derivative term in order to capture the rapid change of the flux and increase the stability region for the time step size [51]. With the exponential transformation, the scalar flux is decomposed as:

$$\phi_g(t,\boldsymbol{r}) = e^{\alpha^{(n)}(t-t_{n-1})}\Phi_g(t,\mathbf{r})\,, \qquad\qquad t_{n-1} \le t \le t_n\,. \tag{2.53}$$

The $\alpha^{(n)}$ is computed at the beginning of each time step by solving the EPKEs, which are presented later in this section, with the estimated reactivity. For the fully-implicit BE discretization, the time derivative term of the scalar flux is rewritten as:

$$\begin{aligned}\frac{\partial\phi_g^{(n)}(\boldsymbol{r})}{\partial t} &= \alpha^{(n)}\phi_g^{(n)}(\boldsymbol{r}) + e^{\alpha^{(n)}(t_n-t_{n-1})}\frac{\partial\Phi_g(t,\boldsymbol{r})}{\partial t}|_{t=t_n} \\ &\approx \alpha^{(n)}\phi_g^{(n)}(\boldsymbol{r}) + e^{\alpha^{(n)}\Delta t_n}\frac{\Phi_g^{(n)}(\boldsymbol{r}) - \Phi_g^{(n-1)}(\boldsymbol{r})}{\Delta t_n} \\ &= \alpha^{(n)}\phi_g^{(n)}(\boldsymbol{r}) + \frac{\phi_g^{(n)}(\boldsymbol{r}) - e^{\alpha^{(n)}\Delta t_n}\phi_g^{(n-1)}(\boldsymbol{r})}{\Delta t_n}\,.\end{aligned} \tag{2.54}$$

The delayed neutron precursor equations are treated using the analytical integration method:

$$C_k^{(n)}(\boldsymbol{r}) = C_k^{(n-1)}(\boldsymbol{r})e^{-\lambda_k^{(n)}(\boldsymbol{r})\Delta t_n} + \int_{t_{n-1}}^{t_n}\beta_k(\boldsymbol{r},t)S_F(\boldsymbol{r},t)e^{\lambda_k^{(n)}(\boldsymbol{r})}dt\,, \tag{2.55}$$

with $\lambda(\boldsymbol{r},t)$ assumed to be constant $\lambda_k^{(n)}(\boldsymbol{r})$ for $t \in [t_{n-1}, t_n]$. A second-order approximation is then made to treat the delayed fission source term $\beta_k(\boldsymbol{r},t)S_F(\boldsymbol{r},t)$ as

$$\begin{gathered}\beta_k(\mathbf{r},t)S_F(\mathbf{r},t) \approx \beta_k^{(n)}(\mathbf{r})S_F^{(n)}(\mathbf{r})\tfrac{\tilde{t}^2+\tilde{t}\gamma\Delta t_n}{(1+\gamma)(\Delta t_n)^2} \\ +\beta_k^{(n-1)}(\mathbf{r})S_F^{(n-1)}(\mathbf{r})\left(1 - \tfrac{\tilde{t}^2+(\gamma-1)\Delta t_n\tilde{t}}{\gamma(\Delta t_n)^2}\right) \\ +\beta_k^{n-2}(\mathbf{r})S_F^{n-2}(\mathbf{r})\tfrac{\tilde{t}^2-\tilde{t}\Delta t_n}{(1+\gamma)\gamma(\Delta t_n)^2}\,, \\ \tilde{t} = t - t_{n-1}\,, \gamma = \tfrac{\Delta t_{n-1}}{\Delta t_n}\,.\end{gathered} \tag{2.56}$$

Then the delayed fission neutron source term can be expressed as

$$S_d^{(n)}(\boldsymbol{r}) = \omega^{(n)}(\boldsymbol{r})S_F^{(n)}(\boldsymbol{r}) + \widetilde{S}_d^{(n-1)}(\boldsymbol{r})\,, \tag{2.57}$$

where $\omega^{(n)}$ is the term to denote the contribution from the fission source at the current time step and $\widetilde{S}_d^{(n-1)}$ is the term related to the delayed neutron precursors and fission source of the previous two time steps. Making use of Eq. (2.52) and Eq. (2.57) reduces the time-dependent Boltzmann transport equation to the following discretized equation

$$\begin{aligned}\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\varphi_g^{(n)}\left(\boldsymbol{r},\boldsymbol{\Omega}\right)+\Sigma_{t,g}^{(n)}(\boldsymbol{r})\varphi_g^{(n)}\left(\boldsymbol{r},\boldsymbol{\Omega}\right)=\frac{\chi_g(\boldsymbol{r})}{4\pi}S_F^{(n)}\left(\boldsymbol{r}\right)+&\\ \sum_{g'=1}^{G}\Sigma_{s0,g'\to g}^{(n)}\left(\boldsymbol{r}\right)\phi_{g'}^{(n)}(\boldsymbol{r})+\frac{S_{tr,g}^{(n)}(\boldsymbol{r})}{4\pi}\,.&\end{aligned} \tag{2.58}$$

Here

$$\chi_g^{(n)}(\boldsymbol{r})=\chi_{p,g}^{(n)}(\boldsymbol{r})\Big[1-\beta^{(n)}(\boldsymbol{r})\Big]+\chi_{d,g}^{(n)}(\boldsymbol{r})\beta^{(n)}(\boldsymbol{r})\,, \tag{2.59}$$

and

$$S_{tr,g}^{(n)}(\boldsymbol{r})=A_g(\boldsymbol{r})\phi_g^{(n)}(\boldsymbol{r})+B_g(\boldsymbol{r})S_F^{(n)}(\boldsymbol{r})+C_g^{(n)}(\boldsymbol{r})\,, \tag{2.60a}$$

$$A_g^{(n)}(\boldsymbol{r})=-\frac{1}{v_g\Delta t_n}(1+\alpha^{(n)}\Delta t_n)\,, \tag{2.60b}$$

$$B_g^{(n)}(\boldsymbol{r})=\chi_{d,g}(\boldsymbol{r})\Big[\omega^{(n)}(\boldsymbol{r})-\beta^{(n)}(\boldsymbol{r})\Big], \tag{2.60c}$$

$$C_g^{(n)}(\boldsymbol{r})=\chi_{d,g}(\boldsymbol{r})\widetilde{S}_d^{(n-1)}(\boldsymbol{r})+\frac{e^{\alpha^{(n)}\Delta t_n}\phi_g^{(n-1)}(\boldsymbol{r})}{v_g\Delta t_n}\,, \tag{2.60d}$$

formulate a fixed source problem denoted as Transient Fixed Source Problem (TFSP). In Eqs. (2.60), A may be viewed as a cross section, whereas B may be viewed like a fission spectrum, and C is the transient source from delayed fission neutrons and time discretization.

2.4.2 Source Iteration and CMFD Acceleration

For this TFSP, the source iteration scheme is also used. In operator form, the TFSP is represented as:

$$\mathcal{L}\boldsymbol{\varphi}=\frac{1}{4\pi}\Big[\mathcal{S}+\boldsymbol{\chi}\mathcal{F}\Big]\boldsymbol{\phi}+\mathcal{Q}_{tr}\,. \tag{2.61a}$$

Here

$$\mathcal{Q}_{tr}=\mathcal{A}\boldsymbol{\phi}+\mathcal{B}\mathcal{F}\boldsymbol{\phi}+\mathcal{C}\ , \tag{2.61b}$$

$\boldsymbol{\phi}$ is the vector of group-wise scalar flux; $\mathcal{S}$ and $\mathcal{A}$ are $G\times G$ matrices; $\mathcal{F}$ is the fission operator, i.e. a row vector; $\boldsymbol{\chi}$ and $\mathcal{B}$ are column vectors are applied to the scalar quantities; $\mathcal{C}$ is the column vector.

The details for the source iteration are shown in Algorithm 6. We use ℓ to index the source iteration, since n has been used to index the time point. And, n and k_{eff} are omitted for brevity.

Algorithm 6 Source iteration scheme for TFSP.

```
Input maximum outer iteration number $L$, inner iteration number $L_1$; tolerance $\epsilon_f$; initial guess $\boldsymbol{\phi}^{(0)}$.
for $\ell = 0, 1, \ldots, L$ do
    let $\boldsymbol{\phi}^{(\ell,0)} \leftarrow \boldsymbol{\phi}^{(\ell)}$, and calculate the fission source
        $\mathcal{Q}^{(\ell)} = \boldsymbol{\chi}\mathcal{F}\boldsymbol{\phi}^{(\ell)} + \mathcal{A}\boldsymbol{\phi}^{(\ell)} + \mathcal{B}\mathcal{F}\boldsymbol{\phi}^{(\ell)} + \mathcal{C}\,.$        (Alg 6.1)
    for $\ell_1 = 0, 1, \ldots, L_1$ do
        Compute the angular flux via transport sweep:
            $\mathcal{L}\boldsymbol{\varphi}^{(\ell,\ell_1+1)} = \frac{1}{4\pi}\left[\mathcal{S}\boldsymbol{\phi}^{(\ell,\ell_1)} + \mathcal{Q}^{(\ell)}\right].$        (Alg 6.2)
        Update the scalar flux
            $\boldsymbol{\phi}^{(\ell,\ell_1+1)} = \mathcal{M}_0\boldsymbol{\varphi}^{(\ell,\ell_1+1)}\,.$        (Alg 6.3)
    end for
    Let $\boldsymbol{\phi}^{(\ell+1)} \leftarrow \phi^{(\ell,L_1)}$.
    if $\left\|\mathcal{F}\boldsymbol{\phi}^{(\ell)} - \mathcal{F}\boldsymbol{\phi}^{(\ell+1)}\right\| < \epsilon_f \left\|\mathcal{F}\boldsymbol{\phi}^{(l+1)}\right\|$ then
        break
    end if
end for
```

Following the same derivation of CMFD acceleration for the k-eigenvalue problem, the neutron balance equation for the TFSP on the coarse mesh is:

$$\begin{aligned}\sum_{m' \in N(m)} \Gamma_{(m,m')} \Bigg[- \tilde{D}^{(n)}_{(m,m'),g}(\Phi^{(n)}_{m',g} - \Phi^{(n)}_{m,g}) + \hat{D}^{(n)}_{(m,m'),g}(\Phi^{(n)}_{m,g} + \Phi^{(n)}_{m',g}) \Bigg] \\ + \left(\Sigma^{(n)}_{t,m,g}\Phi^{(n)}_{m,g} - \sum_{g'=1}^{G} \Sigma^{(n)}_{s0,m,g'\to g}\Phi^{(n)}_{m,g'} \right) V_m = \left(\chi^{(n)}_{m,g} S^{(n)}_{F,m} + S^{(n)}_{tr,m,g} \right) V_m\,, \\ 1 \le g \le G\,, \quad 1 \le m \le M\,. \end{aligned} \tag{2.62}$$

The definition of the coefficients are

$$A_{m,g} = \frac{\int_{\boldsymbol{r}\in C_m} A_g(\boldsymbol{r})\phi^{ts}_g(\boldsymbol{r})dV}{\int_{\boldsymbol{r}\in C_m} \phi^{ts}_g(\boldsymbol{r})dV}\,, \tag{2.63a}$$

$$C_{m,g} = \frac{\int_{\boldsymbol{r}\in C_m} C_g(\boldsymbol{r})dV}{V_m}\,, \tag{2.63b}$$

$$B_{m,g} = \frac{\int_{\boldsymbol{r} \in C_m} B_g(\boldsymbol{r}) S_F^{ts}(\boldsymbol{r}) dV}{\int_{\boldsymbol{r} \in C_m} S_F^{ts}(\boldsymbol{r}) dV} \,. \tag{2.63c}$$

This 3D-CMFD TFSP can be solved efficiently [51, 54] when written as

$$(\mathbf{M} - \mathbf{S} - \boldsymbol{\chi}\mathbf{F} - \mathbf{A} - \mathbf{BF})\boldsymbol{\Phi} = \mathbf{C} \,, \tag{2.64}$$

with $\mathbf{A} \in \mathbb{R}^{MG \times MG}$, $\mathbf{B} \in \mathbb{R}^{MG \times M}$ and $\mathbf{C} \in \mathbb{R}$. The operators are defined in the way as

$$[\mathbf{S}\boldsymbol{\Phi}]_{(m-1)G+g} = \sum_{g'=1}^{G} \Sigma_{s0,m,g' \to g} \Phi_{m,g'} \,, \tag{2.65a}$$

$$[\boldsymbol{\chi}\mathbf{F}\boldsymbol{\Phi}]_m = \chi_{m,g} \sum_{g'=1}^{G} v\Sigma_{f,m,g'} \Phi_{m,g'} \,, \tag{2.65b}$$

$$[\mathbf{A}\boldsymbol{\Phi}]_{(m-1)G+g} = A_{m,g} \Phi_{m,g} \,, \tag{2.65c}$$

$$[\mathbf{BF}\boldsymbol{\Phi}]_{(m-1)G+g} = B_{m,g} \sum_{g'=1}^{G} v\Sigma_{f,m,g'} \Phi_{m,g'} \,, \tag{2.65d}$$

$$[\mathbf{C}]_{(m-1)G+g} = C_{m,g} \,. \tag{2.65e}$$

Eq. (2.62) is a fixed source problem which can be solved directly using a linear solver, rather than an eigenvalue problem as Eq. (2.39) which must be solved via iteration for practical simulations. However, this TFSP is quite large and nearly singular, and the solution is computationally expensive.

Algorithm 7 CMFD acceleration for TFSP.

```
1: Compute $\hat{D}$ using the current estimate of the angular flux in the transport system.
2: Solve the CMFD TFSP problem to obtain the scalar flux on the CMFD spatial grid.
3: Scale the angular flux in the transport problem using the new scalar flux from the CMFD
   problem. Also, use the CMFD scalar flux to generate a new fine-grid fission source.
4: Perform iterations on the transport problem (i.e., transport sweeps). During the sweep, the
   fission source is fixed.
5: Repeat steps 1-4 until convergence.
```

More than a simple accelerator for the transport TFSP, the CMFD solver is also an important component of the TML. It is the intermediate level that connects the transport and EPKE solutions. As illustrated in the next section, the CMFD TFSP is formed on the finer time grid. Its solutions are used to calculate the coefficients for the EPKEs, which in turn provides a solution to correct the CMFD solutions. The CMFD solutions are then used to correct the transport results.

2.4.3 Exact Point Kinetics Equation

Integrating Eq. (2.62) with the initial adjoint flux $\Phi^*_{m,g}(0)$ over space and energy, the multigroup CMFD transient equation is simplified to

$$\frac{dp(t)}{dt} = \frac{\rho(t) - \beta^{eff}(t)}{\Lambda(t)} p(t) + \frac{1}{\Lambda(0)} \sum_k \lambda_k(t) \xi_k(t)\,, \tag{2.66}$$

and

$$\frac{d\xi_k(t)}{dt} = \frac{\Lambda(0)}{\Lambda(t)} \beta_k^{eff}(t) p(t) - \lambda_k(t)\xi_k(t)\,,\ k = 1, 2, \cdots K\,. \tag{2.67}$$

The equations are EPKEs. The initial adjoint flux is used to reduce the error resulting from inaccuracies in the shape function [88].

When the coefficients are not obtained from the latest transport solution, they are referred to as the PKEs. It should be noted that, in this dissertation, the term EPKE is only used to denote the point-kinetics equation obtained with transport solution. Otherwise, the term PKE is used.

β^{eff} is the effective delayed neutron fraction. For brevity, the superscript ^{eff} is omitted for the β shown later in the thesis.

Let the vector $\mathbf{P}(t) = \{p(t), \xi_1(t), \cdots \xi_n(t)\}$, and define

$$\mathbf{E}(t) = \begin{pmatrix} \frac{\rho(t)-\beta(t)}{\Lambda(t)} & \frac{\lambda_1(t)}{\Lambda(0)} & \frac{\lambda_2(t)}{\Lambda(0)} & \cdots & \cdots & \frac{\lambda_K(t)}{\Lambda(0)} \\ & -\lambda_1(t) & & & & \frac{\Lambda(0)}{\Lambda(t)}\beta_1(t) \\ & & -\lambda_2(t) & & & \frac{\Lambda(0)}{\Lambda(t)}\beta_2(t) \\ & & & \ddots & & \vdots \\ & & & \cdots & -\lambda_K(t) & \frac{\Lambda(0)}{\Lambda(t)}\beta_K(t) \end{pmatrix}. \tag{2.68}$$

The system of EPKEs now can be expressed in operator notation as:

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{E}(t)\mathbf{P}(t)\,. \tag{2.69}$$

To solve Eq. (2.69), the BE is adopted. To avoid the matrix inversion, the precursor integration methods such as first-order precursor integration (FP) or second-order precursor integration (SP) are used. Eq. (2.67) is now rewritten as:

$$\frac{d\xi_k(t)e^{\lambda_k^{(n)}(t-t^{(n-1)})}}{dt} = \left[\lambda_k^{(n)}\xi_k(t) + \frac{d\xi_k(t)}{dt}\right] e^{\lambda_k^{(n)}(t-t^{(n-1)})} = \frac{\Lambda(0)}{\Lambda(t)}\beta_k(t)p(t)e^{\lambda_k^{(n)}(t-t^{(n-1)})}\,. \tag{2.70}$$

Only the first-order approximation is introduced here. Here the superscript $^{(n)}$ is used to index the time point, since the subscript $_k$ is used to index the delayed neutron precursor. The derivation of the second-order method can be found in [51].

$\frac{\beta(t)p(t)}{\Lambda(t)}$ in time is approximated by:

$$\frac{\beta_k(t)p(t)}{\Lambda(t)} = \frac{\beta_k^{(n-1)}p^{(n-1)}}{\Lambda^{(n-1)}}(1-\gamma) + \frac{\beta_k^{(n)}p^{(n)}}{\Lambda^{(n)}}\gamma\,,\ \gamma = \frac{t-t^{(n-1)}}{t^{(n)}-t^{(n-1)}}\,. \tag{2.71}$$

Substituting Eq. (2.71) into Eq. (2.70), and integrating Eq. (2.70) in $[t^{(n-1)}, t^{(n)}]$, we have:

$$\xi_k^{(n)} - \xi_k^{(n-1)}e^{-\lambda_k^{(n)}\Delta t_n} = \Lambda_0\left[\frac{\beta_k^{(n-1)}p^{(n-1)}}{\Lambda^{(n-1)}}\frac{\kappa_0(\tilde{\lambda}_k^{(n)})}{\lambda_k^{(n)}} + \left(\frac{\beta_k^{(n)}p^{(n)}}{\Lambda^{(n)}} - \frac{\beta_k^{(n-1)}p^{(n-1)}}{\Lambda^{(n-1)}}\right)\frac{\kappa_1(\tilde{\lambda}_k^{(n)})}{\lambda_k^{(n)}}\right], \tag{2.72}$$

where $\kappa_0(x) = 1 - exp(-x)$, $\kappa_1(x) = \frac{1-\kappa_0(x)}{x}$, $\tilde{\lambda}_k^{(n)} = \lambda_k^{(n)}\Delta t_n$, $\Lambda_0 = \Lambda(0)$. Then

$$\begin{aligned}\frac{1}{\Lambda_0}\sum_{k=1}^{K}\lambda_k^{(n)}\xi_k^{(n)} = {} & \frac{\sum_{k=1}^{K}\lambda_k^{(n)}\xi_k^{(n-1)}e^{-\tilde{\lambda}_k^{(n)}}}{\Lambda_0} + \frac{\sum_{k=1}^{K}\beta_k^{(n-1)}[\kappa_0(\tilde{\lambda}_k^{(n)}) - \kappa_1(\tilde{\lambda}_k^{(n)})]}{\Lambda^{(n-1)}}p^{(n-1)} \\ & + \frac{\sum_{k=1}^{K}\beta_k^{(n)}\kappa_1(\tilde{\lambda}_k^{(n)})}{\Lambda^{(n)}}p^{(n)}\,.\end{aligned} \tag{2.73}$$

Using BE to solve Eq. (2.66), we obtain

$$\frac{p^{(n)} - p^{(n-1)}}{\Delta t_n} = \frac{\rho^{(n)} - \beta^{(n)}}{\Lambda^{(n)}}p^{(n)} + \frac{1}{\Lambda_0}\sum_{k=1}^{K}\lambda_k^{(n)}\xi_k^{(n)}\,. \tag{2.74}$$

Defining

$$\Omega_k^{(n)} = \frac{\beta_k^{(n)}}{\lambda_k^{(n)}}\frac{\Lambda_0}{\Lambda^{(n)}}\kappa_1(\tilde{\lambda}_k^{(n)})\,, \tag{2.75a}$$

$$S^{(n)} = \frac{\sum_{k=1}^{K}\lambda_k^{(n)}\xi_k^{(n-1)}e^{-\tilde{\lambda}_k^{(n)}}}{\Lambda_0} + \frac{\sum_{k=1}^{K}\beta_k^{(n-1)}[\kappa_0(\tilde{\lambda}_k^{(n)}) - \kappa_1(\tilde{\lambda}_k^{(n)})]}{\Lambda^{(n-1)}}p^{(n-1)}\,, \tag{2.75b}$$

$$\tau^{(n)} = \frac{\sum_{k=1}^{K}\lambda_k^{(n)}\Omega_k^{(n)}}{\Lambda_0}\,, \tag{2.75c}$$

and substituting Eq. (2.73) into Eq. (2.74), yields

$$p^{(n)} = \frac{p^{(n-1)} + S^{(n)}\Delta t_n}{1 - \tau\Delta t_n - \frac{\rho^{(n)}-\beta^{(n)}}{\Lambda^{(n)}}\Delta t_n}\,, \tag{2.76a}$$

$$\xi_k^{(n)} = \xi_k^{(n-1)} e^{-\lambda_k^{(n)}\Delta t_n} + \Omega_k^{(n)} p^{(n)} + \frac{\beta_k^{(n-1)} p^{(n-1)} \Lambda_0}{\lambda_k^{(n)} \Lambda^{(n-1)}} [\kappa_0(\tilde{\lambda}_k^{(n)}) - \kappa_1(\tilde{\lambda}_k^{(n)})] \,. \tag{2.76b}$$

2.4.4 Transient Multilevel Scheme

So far, we have shown the numerical methods to solve the time-dependent transport equation, CMFD equation and the EPKEs. These methods are then used in the TML scheme to solve the time-dependent NTE efficiently.

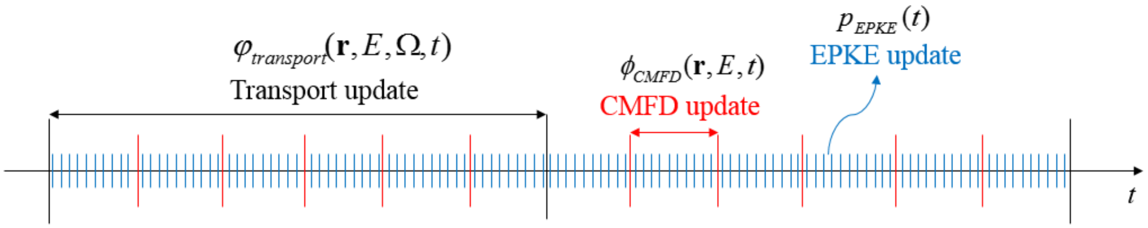

$\varphi_{transport}(\mathbf{r}, E, \Omega, t)$
Transport update
$\phi_{CMFD}(\mathbf{r}, E, t)$
CMFD update
$p_{EPKE}(t)$
EPKE update
t

Figure 2.2: Illustration of the TML Scheme. Courtesy of Ang Zhu [51, 54]. The scheme is a multilevel method composed of three levels–transport, CMFD and EPKE. The solutions for the solver with coarser time steps are corrected by the solutions of the solver with finer time steps

The original TML (TML-3) method is a multilevel PCQM [53] based on the coupling between the transport level and the CMFD level, and the coupling between the CMFD level and the EPKE level.

For the coupling between the transport TFSP and CMFD TFSP, the angular flux in a coarse mesh C_m is factorized as

$$\varphi_g^P(\boldsymbol{r}, \boldsymbol{\Omega}, t) = P_m(t) \Psi_g(\boldsymbol{r}, \boldsymbol{\Omega}, t) \,, \tag{2.77}$$

with the normalization condition

$$\frac{1}{V_m} \int_{\boldsymbol{r} \in C_m} \int_{4\pi} \Psi_g(\boldsymbol{r}, \boldsymbol{\Omega}, t) d\Omega dV = 1 \,. \tag{2.78}$$

V_m is the volume of the whole space. It has been shown that the equation for the coarse-mesh-wise amplitude function, $P_m(t)$, is the same as $\Phi_m(t)$ in the time-dependent CMFD equation [54]. The superscript ^{P} is introduced to indicate that the angular flux solution is predicted.

As a result, in the PCQM scheme, the angular flux is corrected with the CMFD TFSP solution by

$$\varphi_g^C(\boldsymbol{r}, \boldsymbol{\Omega}, t) = \varphi_g^P(\boldsymbol{r}, \boldsymbol{\Omega}, t) \frac{\Phi_{m,g}^C(t)}{\Phi_{m,g}^P(t)}, \boldsymbol{r} \in m \,, \tag{2.79}$$

The Φ^P is homogenized from the transport solution, and Φ^C is the solution of the CMFD TFSP. The CMFD level is solved with finer time steps with the coefficients interpolated between t_{n-1} and t_n.

In the coupling between the CMFD TFSP and EPKEs, the coarse mesh scalar flux is factorized as

$$\Phi_{m,g}(t) = p(t)\Psi_{m,g}(t)\,, \tag{2.80}$$

with the constraints

$$< \Phi^*_{m,g}(0)\frac{\Psi_{m,g}(t)}{v_g} >= C\,. \tag{2.81}$$

Here the operator $< \cdot >$ is defined as

$$< \cdot >= \sum_{m=1}^{M}\sum_{g=1}^{G} V_m(\cdot)\,. \tag{2.82}$$

n' is introduced to index the time point of CMFD level in $[t_{n-1}, t_n]$, so the CMFD scalar flux is corrected by

$$\Phi^C_{m,g}(t_{n'}) = \frac{\Phi^P_{m,g}(t_{n'})p^C(t_{n'})C}{< \Phi^*_{m,g}(0)\frac{\Phi^P_{m,g}(t_{n'})}{v_g} >}\,, \tag{2.83}$$

where $p^C(t_{n'})$ is calculated by the EPKEs using finer time steps. The coefficients in the EPKEs are again interpolated between the coefficients at $t_{n'-1}$ and $t_{n'}$ on the CMFD level.

Parametric studies in [51, 54] show that the TML structure of a 5 ms time step for the transport calculation, 1 ms time step for CMFD, and 0.2 ms time step for EPKEs is reasonable for Light Water Reactor (LWR) super-prompt critical transient simulations. Therefore, in MPACT, using the TML scheme with 5 ms transport time step size, and 5 CMFD steps per transport step, is the default setting. 10 EPKE steps per CMFD step are used as the default to further decrease the error from the EPKEs calculation.

2.4.5 Coupling with Other Physics

For transient analysis of the nuclear reactor, the feedback from the thermal-hydraulics (TH) problem is always taken into consideration. In MPACT, the TH problem is either solved by a simplified internal TH solver [89] or externally by the sub-channel code COBRA-TF (CTF) [90]. For transient simulations performed in the thesis, the simplified TH model is used. In the simplified model, a certain fraction (e.g. 98%) of the total power of a fuel pin is assumed to be deposited homogeneously in the fuel pin. A simple fuel conduction problem is then solved. The ring-wise temperature that determines the effective cross sections generation is then updated. For the transient process, the model performs transient conduction to the fuel rod surface using an adiabatic boundary condition. All the transient problems investigated in this thesis are the super-prompt critical transients. Therefore, the adiabatic boundary condition works well for these transients.

The TH problem and neutronics problem are coupled in an operator splitting way. The neutronics problem is solved by the TML methodology first, and the TH field is updated with the latest pin power distribution. Then the TH field is synchronized with the neutronics solution at the transport time steps. Special attention has been paid to applying the feedback to the TML. On the 3D-CMFD level, the start of the CMFD coefficients is homogenized with the cross section updated by the temperature but using the flux at the end of the time point. The numerical results in [51, 48] show that the current coupling method can achieve reasonable accuracy for solving the super-prompt critical problem with a fixed 5 ms time step. However, the current limitations of the accuracy of TML are now likely the result of the temporal coupling of the multiphysics. One aspect to address in particular would be the proper incorporation of the Doppler effect on reactivity, which influences the amplitude change. In the current scheme, a potential inefficiency of the TML may result by requiring that the transport time step be sufficiently small to accurately capture the Doppler effect which is not explicitly treated (e.g. we do not solve the transient conduction equation) in the coarse levels of the TML framework. Developing an adaptive time-stepping strategy and developing a method for synchronizing the TH and neutronics solutions at the CMFD step [47] are very important to improve the efficiency of the TML scheme.

2.4.6 Flowchart of Transient Multilevel Method

The overall flowchart for TH feedback in the TML algorithm is shown in Figure 2.3. The left vertical blocks represent the general transport transient iteration scheme with TH feedback, where the angular and sub-pin flux shapes are assumed to be accurate. The middle vertical blocks show that the pin-wise amplitude function of the transport solution is corrected using intermediate time steps by performing CMFD steps. The global shape function predicted by the CMFD steps is assumed to be accurate, and the whole-core amplitude is corrected by the fine EPKE steps illustrated in the right-most vertical blocks.

2.4.7 Stability of Time-dependent NTE

The time-dependent problem is an initial value problem and is also a dynamic system. The term "stability" has been introduced to characterize the dynamic behavior of the system. Suppose that we have a simple ordinary differential problem:

$$\frac{d\mathbf{y}}{dt} = f(\mathbf{y}, t)\,, y(0) = y_0\,. \tag{2.84}$$

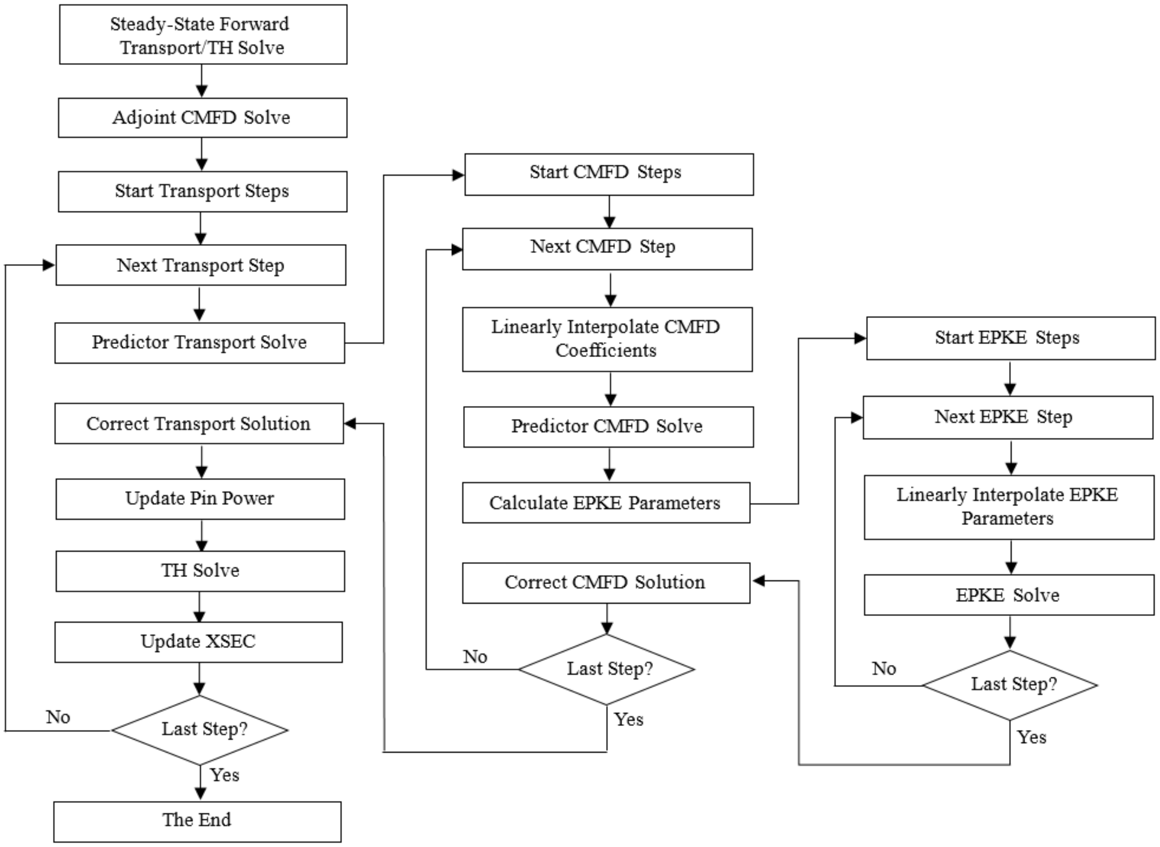


Figure 2.3: Flowchart for TML coupled with TH feedback. Courtesy of Ang Zhu [51, 54].

The solution is stable if for every $\epsilon > 0$, there is a $\delta > 0$ such that when $||\hat{y}_0 - y_0|| < \delta$ then $||\hat{y}(t) - y(t)|| < \epsilon$. Here $\hat{y}(t)$ is the solution of the problem with a perturbed initial value $\hat{y}_0$ [91]. It can be expected that for the time-dependent NTE, when a positive reactivity is inserted and there is no feedback, the problem is unstable. It can also be expected that when the feedback is weak initially, the temporal system is also unstable. A small perturbation can introduce a large difference in the solutions at some time points.

CHAPTER 3

Derivation of Nearly-optimally Partially Converged CMFD

In this chapter, we develop the Fourier analysis for a simplified Coarse Mesh Finite Difference (CMFD)-accelerated neutron transport problem with feedback from flux-dependent cross sections to provide a theoretical explanation, and gain insight into, the observable phenomena in multiphysics calculations with CMFD. These observations are (i) if the acceleration equations are tightly converged every iteration, the overall multiphysics iteration becomes less stable; and (ii) properly loosening the convergence criteria of the acceleration equations at each iteration can stabilize the overall scheme [92, 82, 39].

The Fourier analysis result is used to derive the relationship between relaxation and the partial convergence of the CMFD accelerated transport calculation. Using this relationship we develop a new, more robust multiphysics iteration scheme, nearly-optimally partially converged CMFD (NOPC-CMFD). It is shown that the NOPC-CMFD method in problems with feedback (i) has stability properties comparable to CMFD in problems without feedback, and (ii) requires no relaxation factor. The results presented in this chapter provide a theoretical foundation for the development of a robust multiphysics iteration scheme for nuclear reactor modeling. The implementation of the method and application to various test cases are presented in Chapter 4.

3.1 Theory and Methodology

In this section, the problem of interest, the model problem to be analyzed, and the basis of NDA are presented. The Nonlinear Diffusion Acceleration (NDA) method, rather than the CMFD method, is introduced first because the NDA equations illustrate the process of the iteration scheme, without loss of generality, and they are much simpler to show. The steps to go from NDA to CMFD involve developing a second spatial grid for the low-order problem and associated prolongation and homogenization operators. A complete treatment of this may be found in a number of references [11, 12, 14, 18, 75, 76]. Although the derivation of the CMFD result is not shown, results for CMFD are shown in Section 3.2.

3.1.1 Model Problem

To analyze the stability of the iteration scheme using Fourier analysis, Eq. (2.6) is simplified to a homogeneous, mono-energetic problem in slab geometry with isotropic scattering and reflective boundary conditions.

$$\mu\frac{\partial\psi\left(x,\mu\right)}{\partial x}+\Sigma_t\left(x,\boldsymbol{S}\right)\psi\left(x,\mu\right)=\frac{1}{2}\Big[\Sigma_s\left(x,\boldsymbol{S}\right)+\lambda\nu\Sigma_f\left(x,\boldsymbol{S}\right)\Big]\phi(x)\,, \tag{3.1a}$$

$$\phi\left(x\right)=\int_{-1}^{1}\psi\left(x,\mu\right)d\mu\,, \tag{3.1b}$$

$$\psi\left(0,\mu\right)=\psi(0,-\mu)\,,\qquad 0<\mu\leq 1\,, \tag{3.1c}$$

$$\psi\left(X,\mu\right)=\psi\left(X,-\mu\right)\,,\qquad -1\leq\mu<0\,, \tag{3.1d}$$

$$\frac{1}{X}\int_0^X\int_{-1}^1\psi\left(X,\mu\right)d\mu dx=\Phi_0\,. \tag{3.1e}$$

X is the size of the problem. Reflective boundary conditions are set on both sides of the problem domain. The transport problem of interest is given in Eq. (3.1a). The transport equation, with its boundary conditions, are given in Eq. (3.1a), and Eqs. (3.1c) and (3.1d). The normalization condition is given in Eq. (3.1e).

The mechanism of the feedback physics is generalized by assuming that the macroscopic cross sections are linearly dependent on the scalar flux. The linear model is required for the Fourier analysis. It is written as:

$$\Sigma_i(x,\boldsymbol{S})\approx\Sigma_i(x,\phi)\approx\Sigma_{i0}+\Sigma_{i1}\Big[\phi(x)-\Phi_0\Big]\,,\ i=t,a,s0,f,\dots\,, \tag{3.2}$$

with

$$\Sigma_{i1}=\frac{d\Sigma_i}{d\phi}|_{\phi=\Phi_0}\,. \tag{3.3}$$

The scalar-flux dependence of the macroscopic cross-sections is used as a simple proxy to more complicated multiphysics feedback phenomena, such as temperature-dependent Doppler broadening from a heat conduction and/or fluid dynamics calculation, or nuclide density effects from a fuel-performance or multi-phase fluid dynamics simulation. The linear flux-dependent feedback model is a reasonable representation since the fuel temperature and xenon concentration are dependent on the local pin power, that is mainly determined by the local, energy-integrated, scalar neutron flux. The parameters for the linear-in-scalar-flux macroscopic cross sections are given as

Σ_{i0} and Σ_{i1}, with i representing any of the supported reaction pathways. It should also be noted that whatever the feedback is, the linearization approximation in Eq. (3.2) should be valid as long as the iterative solution is close to the exact solution, where a first-order Taylor expansion is accurate. This approximate model is also supported by the numerical findings that the pin-cell-wise one-group cross sections are nearly linearly dependent on the localized one-group flux [39] in MPACT [4, 65, 93] with a simplified TH model [89]. This leads to an analytic scalar flux solution to the given problem that is constant in space and equal to the average scalar flux normalization value of Φ_0. The normalization factor Φ_0 is used to define the unique eigenfunction $\psi\left(x,\mu\right)$. Therefore, the true solution of this model problem is

$$\lambda = \frac{\Sigma_{a0}}{\nu\Sigma_{f0}}\,, \tag{3.4a}$$

$$\phi\left(x\right) = \Phi_0\,, \tag{3.4b}$$

$$\psi\left(x,\mu\right) = \frac{\Phi_0}{2}\,. \tag{3.4c}$$

For simplicity, the eigenvalue problem in Eq. (3.1a) is rewritten in operator notation,

$$\mathcal{L}\mathit{\Psi} = \frac{1}{2}\Big[\mathcal{S} + \lambda\mathcal{F}\Big]\boldsymbol{\Phi}\,, \tag{3.5}$$

with

$$\boldsymbol{\Phi} = \mathcal{M}_0\mathit{\Psi}\,, \tag{3.6}$$

in which

$$\mathcal{L} = \mu\frac{d}{dx} + \Sigma_t\,, \tag{3.7a}$$

$$\mathcal{S} = \Sigma_s\,, \tag{3.7b}$$

$$\mathcal{F} = \nu\Sigma_f\,, \tag{3.7c}$$

$$\mathcal{M}_i = \int_{-1}^{1}\mu^i(\cdot)d\mu, i = 0, 1\cdots\,. \tag{3.7d}$$

The operators here are strictly the one-group, homogeneous ones, and are not the ones defined in Section 2.3.

3.1.2 Nonlinear Diffusion Acceleration

The variants of NDA utilize the solution of the low-order diffusion problem to accelerate a high-order transport calculation [13]. For continuous cases, the low-order problem can be derived by integrating Eq. (3.1a) over the angular variable to formulate the neutron balance equation as:

$$\frac{dJ\left(x\right)}{dx} + \Sigma_t\left(x\right)\phi\left(x\right) = \Big[\Sigma_s\left(x\right) + \lambda\nu\Sigma_f\left(x\right)\Big]\phi\left(x\right) \ . \tag{3.8}$$

The neutron current J is next approximated by Fick's law with a nonlinear correction, and the derivative of current is rewritten as:

$$\frac{dJ(x)}{dx} = -\frac{d}{dx}D\left(x\right)\frac{d\phi(x)}{dx} + \hat{D}\left(x\right)\phi(x)\,, \tag{3.9}$$

with

$$D(x) = \frac{1}{3\Sigma_t(x)}\,, \tag{3.10a}$$

and

$$\hat{D}\left(x\right) = \Big[\frac{dJ^{TS}(x)}{dx} + \frac{d}{dx}D\left(x\right)\frac{d\phi^{TS}(x)}{dx}\Big]/\phi^{TS}(x)\,, \tag{3.10b}$$

in which, the current $J^{TS}(x)$ and scalar flux $\phi^{TS}(x)$ are transport solutions denoted by ^{TS}, i.e.,

$$J^{TS} = \mathcal{M}_1\varPsi\,. \tag{3.10c}$$

Introducing Eq. (3.9) into Eq. (3.8), the neutron balance equation may be rewritten as a diffusion eigenvalue problem:

$$-\frac{d}{dx}D\left(x\right)\frac{d\phi(x)}{dx} + \Big[\Sigma_a\left(x\right) + \hat{D}\left(x\right)\Big]\phi\left(x\right) = \lambda\nu\Sigma_f\left(x\right)\phi\left(x\right) \ . \tag{3.11}$$

Though there are numerous methods to solve this generalized eigenvalue problem (such as the generalized Davidson method [82], JFNK method [83]), the most popular method in the nuclear reactor analysis field is still the WS power iteration method shown in Algorithm 4.

3.1.2.1 Partial Convergence

One common "mistake" or inconsistency in other analyses of the low-order problem is to assume the eigenvalue problem is fully converged [75, 94, 76, 18]. In practice, this eigenvalue problem is only partially converged. To account for the partial convergence, we introduce the parameter l as an index for the power iteration with the maximum value L. DeCART [2], nTRACER [3], MPACT [93], and likely other CMFD-accelerated deterministic transport codes all make use of L as an input to specify the maximum number of power iterations allowed per outer iteration. Methods reviewed in [78] have been used to determine the WS parameter, λ_s, in problems without feedback. These methods improve the convergence rate of the low-order problem without increasing the computational burden in most cases. However, it is difficult to analyze these methods since

Algorithm 8 NDA iteration scheme for the neutron transport problem with feedback.

```
Input iteration number $N, L$, tolerance $\epsilon_k, \epsilon_f$, initial guess $\mathbf{\Phi}^{(0)}$, $J^{(0)}$, $\lambda^{(0)}$.
for $n = 0, 1, 2, \cdots, N$ do
    Apply the feedback. For the model problem with flux-dependent feedback, the cross section is updated by:
        $\Sigma_i^{(n)}(x) = \Sigma_{i0} + \Sigma_{i1}\Big[\phi^{(n)}(x) - \Phi_0\Big]$ .
    (When the spatial discretization is used, the coarse mesh wise cross sections are homogenized with updated cross sections on the fine mesh of the transport problem.)
    Compute the diffusion coefficient and the nonlinear current correction factor for the low-order diffusion equation with $J^{(n)}$ and $\mathbf{\Phi}^{(n)}$ using Eqs. (3.10).
    Solve the eigenvalue system with $L$ power iterations using Algorithm 4 with initial guess $\mathbf{\Phi}^{(n)}$.
    Update the transport flux and eigenvalue with $\mathbf{\Phi}^{(n+1/2)} \leftarrow \mathbf{\Phi}^{(n,L)}$, $\lambda^{(n+1/2)} \leftarrow \lambda^{(n,L)}$.
    Perform a transport sweep with the solution from the low-order problem:
        $\mathcal{L}^{(n)}\, \Psi^{\left(n+\frac{1}{2}\right)} = \frac{1}{2}\Big[\mathcal{S}^{(n)} + \lambda^{\left(n+\frac{1}{2}\right)}\mathcal{F}^{(n)}\Big]\mathbf{\Phi}^{\left(n+\frac{1}{2}\right)}$ .    (Alg 8.1)
    Update the scalar flux, current, and eigenvalue estimates for the next iteration by:
        $J^{(n+1)} = \Phi_0 \frac{\mathcal{M}_1\, \Psi^{\left(n+\frac{1}{2}\right)}}{\left\|\mathcal{M}_0\, \Psi^{\left(n+\frac{1}{2}\right)}\right\|}$ ,    (Alg 8.2a)
        $\mathbf{\Phi}^{(n+1)} = \Phi_0 \frac{\mathcal{M}_0\, \Psi^{\left(n+\frac{1}{2}\right)}}{\left\|\mathcal{M}_0\, \Psi^{\left(n+\frac{1}{2}\right)}\right\|}$ .    (Alg 8.2b)
    Let $\lambda^{(n+1)} \leftarrow \lambda^{(n+\frac{1}{2})}$, $k_{eff}^{(n+1)} \leftarrow 1/\lambda^{(n+1)}$.
    if $\left|k_{eff}^{(n+1)} - k_{eff}^{(n)}\right| < \epsilon_c$ and $\left\|\mathcal{F}^{(n)}\Phi^{(n)} - \mathcal{F}^{(n+1)}\Phi^{(n+1)}\right\| < \epsilon_f \left\|\mathcal{F}^{(n+1)}\Phi^{(n+1)}\right\|$ then
        break
    end if
end for
```

they are dependent on iterative solutions. To simplify the analysis in this study, λ_s is assumed to be proportional to the eigenvalue with a fixed WS ratio r, i.e:

$$r = \frac{\lambda_s}{\lambda}\,. \tag{3.12}$$

In practice, the partial convergence of the low-order diffusion solution is determined by the aggressiveness of the WS parameter λ_s and the power iteration number, L. *Therefore, to study the effect of the partial convergence of the low-order problem is to study the effect of the power iteration number L and the WS ratio r.*

3.1.3 Picard Iteration for NDA with Feedback

The overall iteration scheme for the NDA is shown in Algorithm 8. The scheme is a simplified version of the scheme shown in Figure 2.1.

We are aware that in practice there may be an inner loop to solve Eq. (Alg 8.1) for converging the scattering source. However, to reduce the number of parameters to be investigated in the analysis presented in this chapter, it is assumed that only one transport sweep is performed with no inner iterations on the scattering source.

3.2 Fourier Analysis

The process of Fourier analysis starts with the linearization of the equations comprising the iteration scheme, and then introduces the Fourier Ansatz, that is a guess for the representative form of the general solution in terms of its Fourier modes. In this section we show the analysis procedure of NDA while that of CMFD is shown in Appendix A.2.

3.2.1 Linearization

To perform the linearization, it is assumed that the solution estimates of the n^{th} outer iteration are close to the true solution. This is equivalent to expanding the terms in the equations of Algorithm 8 in terms of the exact solution, and a small error term dependent on the iteration index. Substituting this expansion yields:

$$\psi^{\left(n+\frac{1}{2}\right)}(x,\mu) = \frac{\Phi_0}{2} + \epsilon\psi_1^{\left(n+\frac{1}{2}\right)}(x,\mu) + O(\epsilon^2)\,, \tag{3.13a}$$

$$\phi^{(n)}(x) = \Phi_0 + \epsilon\phi_1^{(n)}(x) + O(\epsilon^2)\,, \tag{3.13b}$$

$$\phi^{\left(n+\frac{1}{2}\right)}(x) = \Phi_0 + \epsilon\phi_1^{\left(n+\frac{1}{2}\right)}(x) + O(\epsilon^2)\,, \tag{3.13c}$$

$$J^{(n)}(x) = \epsilon J_1^{(n)}(x) + O(\epsilon^2)\,, \tag{3.13d}$$

$$\lambda^{\left(n+\frac{1}{2}\right)} = \lambda_0 + \epsilon\lambda_1^{\left(n+\frac{1}{2}\right)} + O(\epsilon^2)\,. \tag{3.13e}$$

Then the cross sections updated with the linear flux-dependent model are expanded as:

$$\Sigma_i^{(n)}(x) = \Sigma_{i0} + \epsilon\Sigma_{i1}\phi_1^{(n)}(x) + O(\epsilon^2)\,. \tag{3.14}$$

During the WS power iteration, the iterates are linearized in a similar way as:

$$\phi^{(n,l)}(x) = \Phi_0 + \epsilon\phi_1^{(n,l)}(x) + O(\epsilon^2)\,, \tag{3.15a}$$

$$\lambda^{(n,l)} = \lambda_0 + \epsilon\lambda_1^{(n,l)} + O(\epsilon^2)\,, \tag{3.15b}$$

$$\lambda_s^{(n,l)} = r\lambda_0 + r\epsilon\lambda_1^{(n,l)} + O(\epsilon^2)\,. \tag{3.15c}$$

Substituting Eqs. (3.13) and (3.14) into the equations involved in the process of NDA and neglecting the $O(\epsilon^2)$ terms gives the relations for the propagation of the leading order errors through the iteration scheme. This result is given in Eqs. (3.16).

$$\phi_1^{(n,0)}(x) = \phi_1^{(n)}(x)\,, \tag{3.16a}$$

$$\begin{aligned}\frac{1}{3\Sigma_{t0}}\frac{d^2\phi_1^{(n,l+1)}(x)}{dx^2} &- \Big(1-r\Big)\Sigma_{a0}\phi_1^{(n,l+1)}(x)\\ &= \frac{dJ_1^{(n)}(x)}{dx} + \frac{1}{3\Sigma_{t0}}\frac{d^2\phi_1^{(n)}(x)}{dx^2} + \Sigma_{t0}\gamma\phi_1^{(n)}(x) - \Big(1-r\Big)\Sigma_{a0}\phi_1^{(n,l)}(x)\,,\end{aligned} \tag{3.16b}$$

$$\phi_1^{\left(n+\frac{1}{2}\right)}(x) = \phi_1^{(n,L)}(x)\,, \tag{3.16c}$$

$$\mu\frac{\partial\psi_1^{\left(n+\frac{1}{2}\right)}(x,\mu)}{\partial x} + \Sigma_{t0}\psi_1^{\left(n+\frac{1}{2}\right)}(x,\mu) = \frac{1}{2}\Sigma_{t0}\Big[\phi_1^{\left(n+\frac{1}{2}\right)}(x) - \gamma\phi_1^{(n)}(x)\Big]\,, \tag{3.16d}$$

$$\phi_1^{(n+1)} = \mathcal{M}_0\psi_1^{\left(n+\frac{1}{2}\right)}\,, \tag{3.16e}$$

$$J_1^{(n+1)} = \mathcal{M}_1\psi_1^{\left(n+\frac{1}{2}\right)}\,, \tag{3.16f}$$

$$\lambda_1 = 0\,, \tag{3.16g}$$

where

$$\gamma = \left(\frac{\Sigma_{a1}}{\Sigma_{a0}} - \frac{\Sigma_{f1}}{\Sigma_{f0}}\right)\frac{\Sigma_{a0}}{\Sigma_{t0}}\Phi_0\,. \tag{3.16h}$$

A detailed discussion about γ is given in [39]. In summary, the parameter γ notionally represents the intensity or magnitude of the cross section feedback due to the perturbation of the feedback. A positive γ indicates that the system has negative feedback, which is a key factor for having a physically stable reactor. In this thesis, γ is referred to as the *feedback intensity*.

The typical γ for PWRs is on the order of 10^{-4}, with a maximum γ less than 2×10^{-3}. We also note that γ may be treated as a local quantity, though it is expressed here with Φ_0. Further discussion on this point is given in Chapter 4. The expression used here in Eq. (3.16h) is to present a way to calculate the feedback intensity that we will make use of in Chapter 4.

After simplification, it may be found that $\lambda_1 = 0$, indicating that the feedback analyzed for the iteration scheme should be independent of the global parameter, k. Therefore, the feedback effect from the critical boron search will not be explicitly analyzed.

3.2.2 Fourier Ansatz

Now the Fourier Ansatz is introduced by assuming:

$$\phi_1^{(n)}(x) = \theta^n \beta_1 e^{i\omega\Sigma_{t0}x}\,, \tag{3.17a}$$

$$J_1^{(n)}(x) = \theta^n i\beta_2 e^{i\omega\Sigma_{t0}x}\,, \tag{3.17b}$$

$$\phi_1^{(n,l)}(x) = \theta^n \alpha^{(l)} e^{i\omega\Sigma_{t0}x}\,, \tag{3.17c}$$

$$\phi_1^{(n+\frac{1}{2})}(x) = \theta^n \beta_3 e^{i\omega\Sigma_{t0}x}\,, \tag{3.17d}$$

$$\psi_1^{(n)}(x,\mu) = \theta^n a(\mu) e^{i\omega\Sigma_{t0}x}\,. \tag{3.17e}$$

θ is the iterative eigenvalue, ω is the Fourier frequency, and βs, a and α are the magnitudes of the error at the very first iteration. Eqs. (3.17) are substituted into Eqs. (3.16) for the errors to obtain θ as a function of ω, the frequency-dependent eigenvalues of the iterative method (or iteration matrix).

$$\begin{aligned}\Big[-\frac{\omega^2}{3} - (1-r)\,(1-c)\Big]\Sigma_{t0}\alpha^{(l+1)} &= \\ &\Big[-\omega\beta_2 - \frac{\omega^2}{3}\beta_1 + \gamma\beta_1 - (1-r)\,(1-c)\,\alpha^{(l)}\Big]\Sigma_{t0}\,, \end{aligned} \tag{3.18a}$$

$$\alpha^{(0)} = \beta_1\,, \tag{3.18b}$$

$$\beta_3 = \alpha^{(L)}\,, \tag{3.18c}$$

$$\theta\beta_1 = \frac{\arctan\left(\omega\right)}{\omega}\left(\beta_3 - \gamma\beta_1\right) \ , \tag{3.18d}$$

$$\theta\beta_2 = -\frac{1}{\omega}\left[1 - \frac{\arctan\left(\omega\right)}{\omega}\right]\left(\beta_3 - \gamma\beta_1\right) \ . \tag{3.18e}$$

3.2.3 Fourier Analysis Result

The Fourier analysis results consist of the final expressions of the spectral radius for both the partially converged NDA and partially converged CMFD.

3.2.3.1 NDA

After simplification of Eqs. (3.18), the iterative eigenvalue θ of the error with frequency ω is

$$\theta\left(\omega\right) = \left[\Lambda^L\left(\omega\right) - \gamma\right] f_{TS}\left(\omega\right) + \left[1 - \Lambda^L\left(\omega\right)\right]\left[f_{NDA}\left(\omega\right) - \frac{3\gamma}{\omega^2} f_{TS}\left(\omega\right)\right] , \tag{3.19}$$

with

$$f_{TS}\left(\omega\right) = \frac{\arctan\left(\omega\right)}{\omega} \ , \tag{3.20a}$$

$$f_{NDA}\left(\omega\right) = (1 + \frac{3}{\omega^2}) f_{TS}\left(\omega\right) - \frac{3}{\omega^2} \ , \tag{3.20b}$$

$$c' = 1 - (1 - r)(1 - c) \ , \tag{3.20c}$$

$$\Lambda\left(\omega\right) = \frac{1 - c'}{1 - c' + \frac{\omega^2}{3}} \ . \tag{3.20d}$$

f_{TS} is the iterative eigenvalue for the source iteration, and f_{NDA} is the iterative eigenvalue for the classical NDA method. The phrase **classical CMFD** refers to the NDA/CMFD-accelerated scheme, where the CMFD solution is fully converged. c' is the effective scattering ratio, and Λ is the error decay rate per power iteration.

The spectral radius is then given by:

$$\rho = \max_{j>0}\left|\theta(\omega_j)\right| , \tag{3.21}$$

with the allowable Fourier modes being of the following discrete form to satisfy the reflective boundary conditions:

$$\omega_j = j\frac{\pi}{\Sigma_{t0}X} \ , \ \ j = 1, 2, \cdots \ . \tag{3.22}$$

The range of the parameters involved in this analysis is specific to the PWR problem and given in Table 3.1.

Table 3.1: Range of parameters used in Fourier analysis.

Symbol	c	γ	r	$\Sigma_t X$
Definition	Scattering Ratio	Feedback intensity	WS ratio	Problem Size (mfp)
Range	< 0.96	$[0.00025, 0.002]$	$[0, 1)$	$[0, 500]$

3.2.3.2 CMFD

The spectral radius for CMFD is presented in this section, with the detailed derivation presented in Appendix A.2. When the coarse cell has the width Δ and contains q evenly-spaced fine mesh cells, the error transition matrix $\mathbf{T}(\omega)$ has the following definition:

$$\mathbf{T}(\omega) = \tilde{\mathbf{H}}(\omega)(1-\gamma) - \Big[1 - \Lambda^L(\omega)\Big]\mathbf{u}\frac{3\Sigma_t\Delta(e^{i\Sigma_t\Delta\omega} - 1)\tilde{\mathbf{G}}(\omega) + \gamma 3(\Sigma_t\Delta)^2\frac{\mathbf{1}^T}{q}\tilde{\mathbf{H}}(\omega)}{2 - 2cos\left(\Sigma_t\Delta\omega\right)}\,, \tag{3.23}$$

with

$$\Lambda(\omega) = \frac{1-c'}{1-c'+\frac{2-2cos(\Sigma_t\Delta\omega)}{3(\Sigma_t\Delta)^2}}\,, \tag{3.24a}$$

$$\tilde{\mathbf{H}} \in \mathbb{C}^{q\times q},\ \tilde{\mathbf{G}}^T \in \mathbb{C}^q,\ \mathbf{u} \in \mathbb{C}^q,\ \mathbf{1} \in \mathbb{C}^q\,. \tag{3.24b}$$

$\tilde{\mathbf{H}}$ is the error transition matrix for the fine mesh scalar flux due to the discretized transport sweep operator. $\tilde{\mathbf{G}}$ is the operator for the coarse mesh surface net current obtained from the fine mesh surface angular flux. The definition of these two terms can be found in [16, 94]. A more recent and thorough discussion of these terms is presented in [76]. The term $\mathbf{u}$ is the spatial prolongation operator for the coarse mesh to fine mesh scalar flux. In [94], β is used for this operator. However, In this chapter we use β to represent relaxation. Unless specified otherwise, the flat prolongation operator of classical CMFD is used, and $\mathbf{u}$ is $\mathbf{1}$. The incorporation of linear (or higher order) spatial prolongation operators is easily handled through modification of $\mathbf{u}$, and in general affects the spectral radius in a fundamentally different way than the inclusion of feedback. The results obtained here and corresponding discussions with respect to feedback and partial convergence are expected to hold true for the CMFD variants discussed in [94]–including linear prolongation.

The iterative eigenvalue from the coarse mesh operator for the Fourier mode with frequency ω_j is:

$$\theta(\omega_j) = \max\left|\mathbf{Eigs}\Big(\mathbf{T}(\omega_j)\Big)\right|\,, \tag{3.25}$$

with the allowable Fourier frequencies:

$$\omega_j = j\frac{\pi}{\Sigma_{t0}\Delta J}\,, \quad j = 1, 2, \cdots, J-1\,, \tag{3.26}$$

where J is the number of coarse meshes in the model problem. ω_j possesses a limit range due to aliasing:

$$\mathbf{T}(\omega_j) = \mathbf{T}(\omega_{j-2J}) = \overline{\mathbf{T}}(\omega_{2J-j})\,. \tag{3.27}$$

The spectral radius is again given by the maximum of all the iterative eigenvalues over the frequency shown by Eq. (3.21).

3.2.3.3 Fully Converged NDA/CMFD with Flux Relaxation

The expression of the spectral radius of the fully converged NDA/CMFD with flux under-relaxation is presented in this section. The NDA with flux relaxation is given by Algorithm 3 in [39]. The flux relaxation is different from the pin power relaxation, that is given in Algorithm 2 in [39] and adopted in most practical simulation codes. It has been shown in [39] that the NDA with flux relaxation can be more stable than NDA with pin power relaxation when the problem domain optical thickness is larger than 50 mean free path (mfp).

The flux relaxation factor β is applied to update $\mathbf{\Phi}^{\left(n+\frac{1}{2}\right)}$ by

$$\mathbf{\Phi}^{\left(n+\frac{1}{2}\right)} = \beta\mathbf{\Phi}^{\left(n+\frac{1}{2}\right)} + (1-\beta)\mathbf{\Phi}^{(n)}\,, \tag{3.28}$$

when solving Eq. (Alg 8.1). The iterative eigenvalue of the Fourier modes is written as:

$$\theta\left(\omega\right) = \begin{cases} \left(1-\beta-\gamma\right) f_{TS}\left(\omega\right) + \beta\left[f_{NDA}\left(\omega\right) - \frac{3\gamma}{\omega^2} f_{TS}\left(\omega\right)\right], & \text{NDA, (Algorithm 3) in [39]}\,, \\ \max\left|\mathbf{Eigs}\Big(T(\omega)\Big)\right|, & \text{CMFD}\,, \end{cases} \tag{3.29a}$$

with

$$\mathbf{T}\left(\omega\right) = \tilde{\mathbf{H}}\left(\omega\right)\left(1-\gamma\right) - \beta\mathbf{u}\frac{3\Sigma_t\Delta(e^{i\Sigma_t\Delta\omega}-1)\tilde{\mathbf{G}}\left(\omega\right) + \gamma 3(\Sigma_t\Delta)^2\frac{\mathbf{1}^T}{q}\tilde{\mathbf{H}}\left(\omega\right)}{2-2cos\left(\Sigma_t\Delta\omega\right)}\,. \tag{3.29b}$$

The detailed derivation process for the spectral radius of CMFD is shown in Appendix A.2 and omitted here for brevity. It should be noted that for the transport sweep, step-characteristics is assumed to be used, so the results can be more instructive for applications with the MOC solver.

3.2.4 Validation of Fourier Analysis Results

The expressions are first verified through careful examination of Eqs. (3.19) to (3.23) and their limits.

- Λ^L is the term denoting the effect of the partial convergence of low-order diffusion solutions. With L and r increasing, the partial convergence term should decrease continuously and vanish eventually when the NDA has been fully converged. This is verified by taking the limits:

  $$\lim_{L\to\infty} \Lambda^L(\omega) = 0\,, \tag{3.30}$$

  and

  $$\lim_{r\to 1} \Lambda^L(\omega) = 0^L = 0\,, \tag{3.31}$$

  because

  $$\Lambda(\omega) = \frac{1-c'}{1-c'+\frac{\omega^2}{3}} \in [0,1)\,.$$

- When γ is 0, i.e. no feedback is present, with both $L\to\infty$ and $r\to 1$, the expression for the iteration eigenvalue $\theta(\omega)$ in the continuous problem is simplified:

  $$\theta(\omega) = (1+\frac{3}{\omega^2})f_{TS}(\omega) - \frac{3}{\omega^2}\,, \tag{3.32}$$

  which is just the expression of θ for the classical NDA. When γ is not zero with both $L\to\infty$ and $r\to 1$, the expression is simplified as

  $$\theta(\omega) = f_{NDA}(\omega) - \gamma(1+\frac{3}{\omega^2})f_{TS}(\omega)\ , \tag{3.33}$$

  which is the expression for the spectral radius of the Algorithm 2 or 3 analyzed in [39] without relaxation.

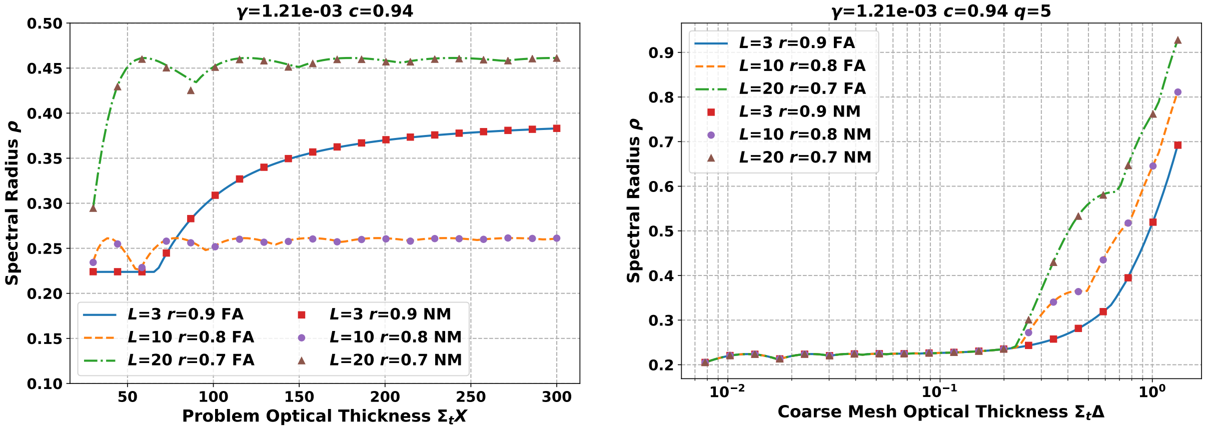
- For the discretized problem, when no feedback is present and the CMFD problem is fully converged, the error transition matrix $\mathbf{T}(\omega)$ in Eq. (3.23) is simplified as

  $$\mathbf{T}(\omega) = \tilde{\mathbf{H}}(\omega) - \mathbf{1}\frac{3\Sigma_t\Delta(e^{i\Sigma_t\Delta\omega}-1)\tilde{\mathbf{G}}(\omega)}{2-2cos(\Sigma_t\Delta\omega)}\,, \tag{3.34}$$

  which is the same error transition matrix of classical CMFD as defined in [16].

To validate the Fourier analysis results, the spectral radii were measured from the numerical simulations of a 1D test code using the model problem and compared to the values predicted by

Eq. (3.21) for NDA and CMFD with partial convergence from various combinations of power iteration number L and WS ratio r. All simulations for this chapter were run with a Gauss–Legendre quadrature set of order 16. The transport discretization was S_N with step-characteristics. For the continuous case, the problem was discretized into 4000 spatial cells, while for the discretized problem, 100 coarse mesh cells were used with 5 fine cells per coarse cell. The feedback intensity γ adopted was 0.00121 (taken from [39]), and the scattering ratio c used was 0.94.



(a) **(b)**

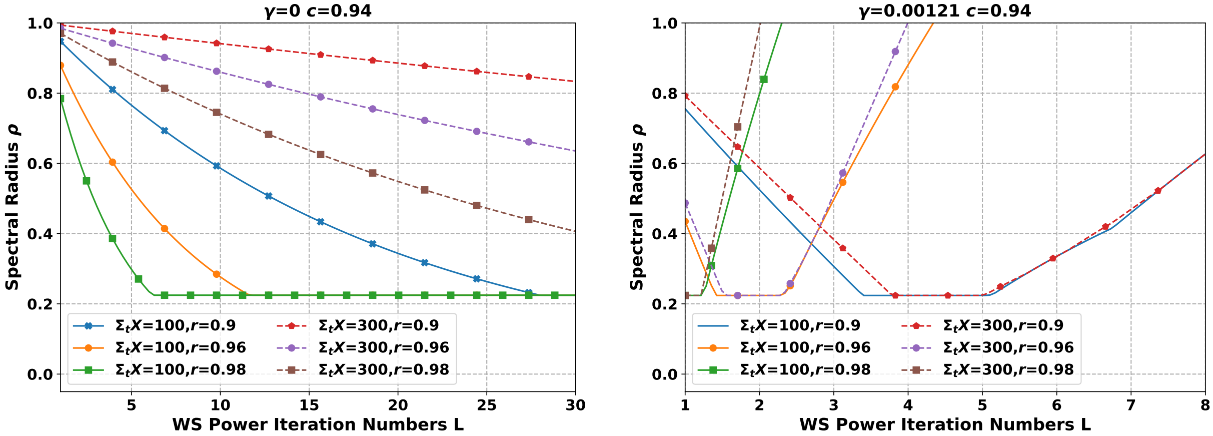
Figure 3.1: Comparison between Fourier analysis and numerical results. **FA** stands for Fourier analysis and **NM** stands for Numerical. (a) shows the results of the NDA in continuous problem. (b) illustrates the spectral radii of CMFD in the spatially discretized problem.

The spectral radii plotted in Figures 3.1 show that the Fourier analysis results and the numerical results from the 1D test code are in very good agreement. There are some outliers for which the spectral radius measured from the numerical simulation is smaller than that from the Fourier analysis. This is likely due to the fast error modes in the numerical simulations. The differences between the spectral radii estimated from numerical simulation and Fourier analysis are smaller than 1%. This verifies the Fourier analysis results. Therefore, we conclude that the Fourier analysis results presented in this section provide a way to analyze the effect of the partial convergence of the low-order diffusion problem of NDA or CMFD on the stability of the iteration scheme for multiphysics simulations.

Though Fourier analysis is a semi-quantitative approach to analyze the stability of the iterative methods, many applications of Fourier analysis have shown that the behavior of real problems is well predicted by the Fourier analysis [74, 16]. It will also be shown in Chapter 4 that what is predicted for the partially converged NDA/CMFD with linearly flux-dependent cross sections in this chapter is consistent with what is observed in the simulation of realistic problems using MPACT.

3.2.5 Effect of Partial Convergence

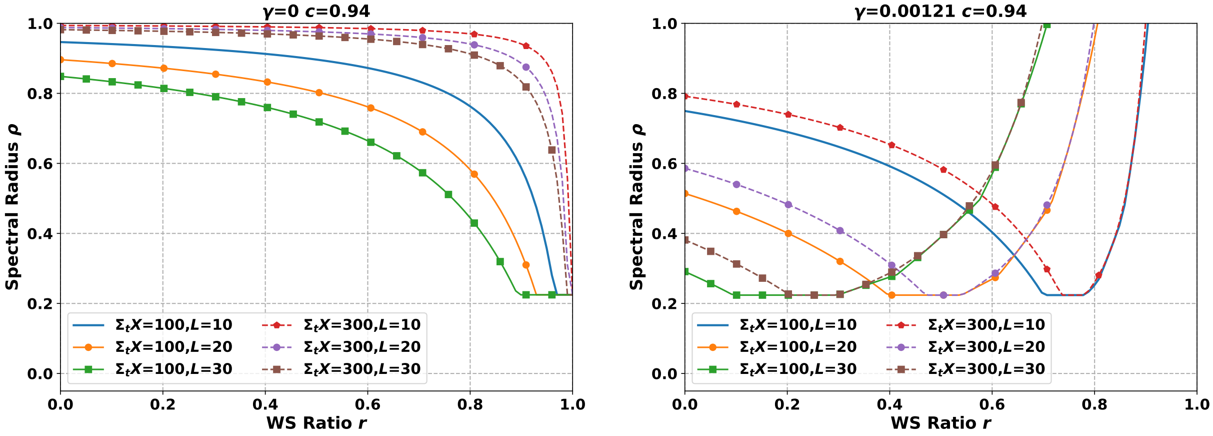
The plots shown in Figures 3.2 to 3.3 show the effect of the partial convergence of low-order diffusion solutions on the stability of the overall acceleration scheme. Several observations are made from these figures:



(a)

(b)

Figure 3.2: Spectral radius as a function of power iteration Number L. (a) shows how L affects the spectral radius in the problem without feedback. (b) shows how L affects the spectral radius in the problem with feedback.



(a)

(b)

Figure 3.3: Spectral radius as a function of the WS ratio r. (a) shows how r affects the spectral radius in the problem without feedback. (b) shows how r affects the spectral radius in the problem with feedback.

- In Figure 3.2a and Figure 3.3a the spectral radius of the iteration scheme decreases with L or r increasing. *This illustrates that a more tightly converged low-order problem is desirable* *for simulations without feedback to achieve optimal convergence rates*. Therefore, fully converged NDA provides a theoretically optimal algorithm *for the case without feedback*.

- When the domain size is large (e.g., $\Sigma_t X = 300$), the power iteration number L has to be larger than 30 or the WS ratio r has to be very close to 1 to achieve the theoretically optimal spectral radius of NDA $\rho = 0.2247$. The results imply the conventional WS power iteration may not be an effective method since a very large L or a near-unit r makes the NDA much more computationally expensive. This is likely why recent research has focused on developing or applying more advanced methods such as the multilevel method [32] or Generalized Davidson method [82], respectively, to solve the low-order diffusion eigenvalue problem more efficiently.

- Figure 3.2b and Figure 3.3b illustrate the effect of the partial convergence for problems with feedback. *If the low-order solutions are too tightly converged, the iteration scheme becomes unstable*. Under this scenario, as shown in Figure 3.2b, loosening the convergence criteria by reducing the power iteration number L or aggressiveness of the WS makes the scheme more stable. These results agree with what has been observed in [32]. Moreover, it can be seen that (i) an optimal convergence rate exists for some combinations of L and r; (ii) the optimal combination is problem-dependent. Therefore, the partial convergence of the low-order problem should be within a specific, problem-dependent range to enable NDA to achieve a fast rate of convergence. It is also observed that when the low-order equation is solved to the same convergence criteria for two problems that are identical in all regards–except one possesses feedback and the other does not–the iteration scheme may perform better in the problem with feedback.

- The plots illustrated in Figure 3.2b and Figure 3.3b are qualitatively similar to plots illustrating the effect of relaxation shown in [39] implying the effect of partial convergence should be interpretable from the perspective of relaxation. The looseness/tightness of the convergence of the low-order problem can induce too much or little under-relaxation.

3.2.6 Partial Convergence vs Relaxation

The link between partially converging the low-order diffusion problem of NDA/CMFD and relaxation arises from the structure of the power iteration. The initial guess of the power iteration solution is the scalar flux from the previous iteration step $\mathbf{\Phi}^{(n)}$ and the solution converges to the exact solution of the low-order equation $\mathbf{\Phi}_f^{\left(n+\frac{1}{2}\right)}$ (f denotes fully-converged) as $L \to \infty$ and $r \to 1$. When the low-order equation is not fully converged, the scalar flux on the right-side of

Eq. (Alg 8.1) is just an intermediate value between $\mathbf{\Phi}^{(n)}$ and $\mathbf{\Phi}_f^{\left(n+\frac{1}{2}\right)}$. This is analogous to the formulation of relaxation that uses a linear combination of $\mathbf{\Phi}^{(n)}$ and $\mathbf{\Phi}_f^{\left(n+\frac{1}{2}\right)}$.

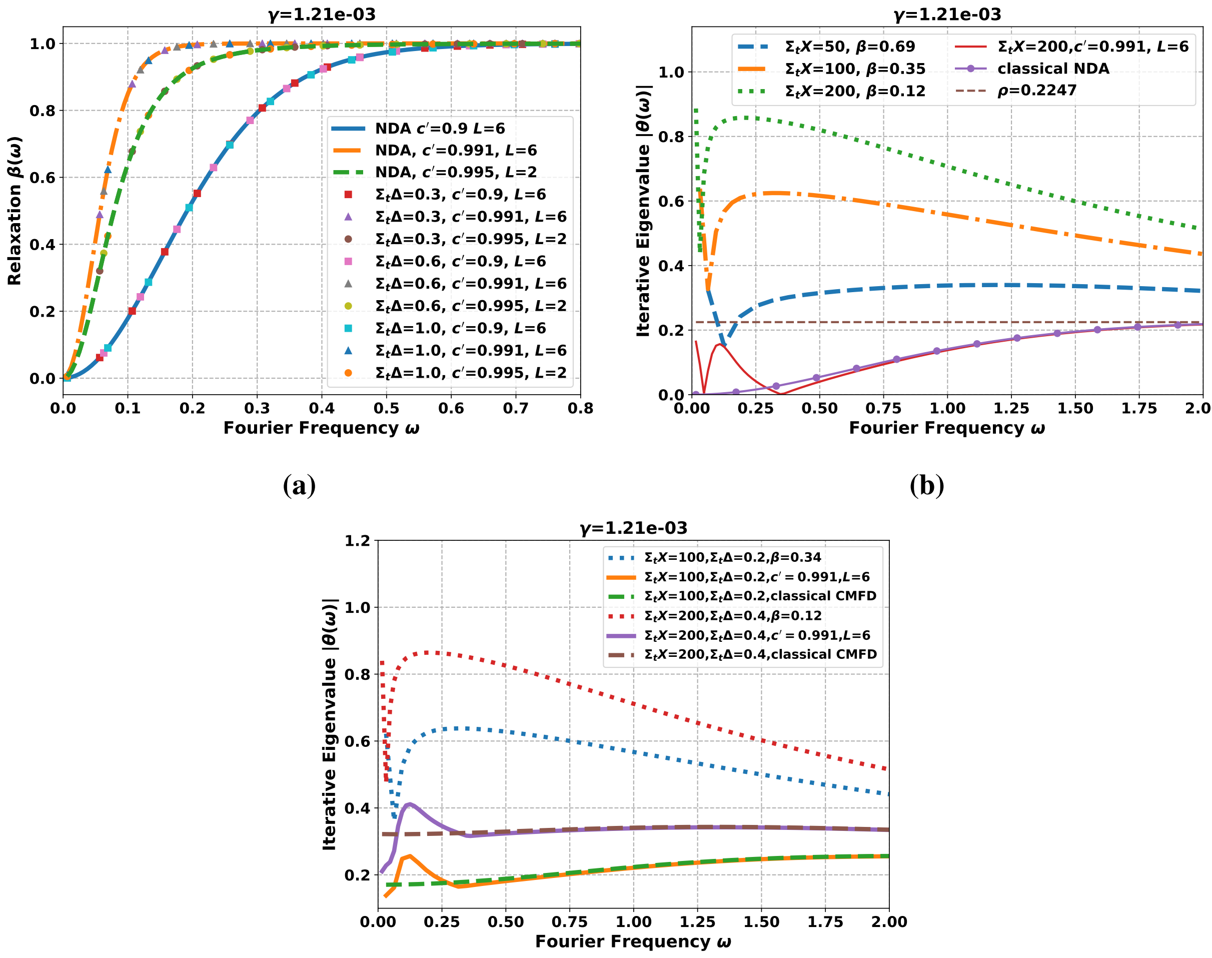
The partial convergence can be shown to be equivalent to a very special under-relaxation, and this relationship can be expressed mathematically. Comparing Eq. (3.19) and Eq. (3.23) with Eqs. (3.29), it can be found that these two sets of equations have the exact same form, with β in Eqs. (3.29) defined as:

$$\beta\left(\omega\right) = 1 - \Lambda^L\left(\omega\right) \, . \tag{3.35}$$

The implicit under-relaxation due to the partial convergence is slightly different from the common under-relaxation. In Eqs. (3.29), the relaxation factor is a constant with respect to the Fourier frequency. Conversely, as Eq. (3.35) illustrates, the under-relaxation from the partial convergence of the low-order equations is ω-dependent.

Figure 3.4a shows how $\beta\left(\omega\right)$ varies with the Fourier frequency for both the spatially continuous and discretized cases. Here, it is observed that $\beta(\omega)$ monotonically increases from 0 to 1 with ω increasing. The results show that the under-relaxation from partial convergence is mainly imposed on the relatively flat error modes. This observation helps explain why the partial convergence of the low-order diffusion solution can stabilize the iterations of problems with feedback, since the flat error modes are shown to be the source of instability in [39], and under-relaxation must be applied to reduce the spectral radius. The partial convergence mainly relaxes the relatively flat error modes and does not affect the high-frequency error modes. This follows from the power iteration procedure as fixed-point iterations generally eliminate the high-frequency modes very quickly. Therefore, when the partial convergence of low-order problem is used, the instability from the relatively flat error modes will be suppressed without altering (or amplifying) the high-frequency error modes.

Figure 3.4b compares the effect of a constant relaxation factor and the effect of the implicit ω-dependent relaxation arising from the partial convergence in the continuous problem. In the figure, the constant relaxation factors are the optimal relaxation factors that make the spectral radius of NDA/CMFD the smallest. It can be observed that even with the optimal relaxation factors, the NDA cannot converge at the same rate as classical NDA without feedback. The reason is that the relaxation factor that reduces the magnitude of the iterative eigenvalues of the relatively flat error modes amplifies the high-frequency error modes. As a result, the spectral radius inevitably becomes larger than the spectral radius of classical NDA. This degradation becomes more severe as the model problem size increases because the relaxation factor β must be closer to 0 to dampen the flat error modes. Under these circumstances $\theta(\omega) \approx f_{TS}(\omega)$, for which the spectral radius approaches 1.



(a)

(b)

(c)

Figure 3.4: $\beta(\omega)$ and $|\theta(\omega)|$ as function of Fourier frequency. (a) shows how relaxation factor $\beta\,(\omega)$ varies with ω for both NDA and CMFD. (b) compares the iterative eigenvalues of error modes for fully converged NDA with optimal constant flux relaxation factor and NDA with partially converged low-order diffusion solutions. (c) compares the iterative eigenvalues of different error modes for fully converged CMFD with optimal constant flux relaxation factor and CMFD with partially converged low-order solutions.

For the discretized problem, as Figure 3.4c shows, CMFD with partial convergence cannot achieve the same spectral radius as the classical CMFD. However, the difference is small, and the spectral radius is still smaller than that of CMFD with an optimal relaxation factor.

The concept of an ω-dependent relaxation factor is somewhat strange. In practice, this may be straightforward to implement for spectral methods, but it does not have an obvious analog for the iteration schemes described here. However, we may understand the ω-dependent relaxation factor as being like a multi-level operator, rather than applying a fixed parameter, where an iteration is performed on a different grid that targets a specific Fourier frequency.

3.3 Relaxation-free Iteration Scheme

The previous section showed that the partial convergence of the low-order problem in NDA/CMFD introduces the notion of a Fourier frequency-dependent relaxation. This behavior makes it possible to develop a robust relaxation-free method based on the nearly-optimal partial convergence of the low-order equations. Consequently, the spectral radius of the classical NDA/CMFD schemes can be achieved for NDA/CMFD in problems with feedback. The effect of partial convergence on the stability of CMFD is close to that of NDA. Therefore, in this section, we first develop the nearly-optimal relaxation-free method to partially converge the low-order diffusion problem of NDA, and then apply the algorithm directly to CMFD.

3.3.1 NOPC Algorithm

In NDA the source of the instability for problems with feedback arises from the flat error modes. As a result, the nearly-optimal partial convergence should make the $\left|\theta\left(\omega_p\right)\right|$ as small as possible. Here, ω_p is a relatively flat Fourier mode ($\omega_p \approx 0$) that is used to predict the nearly-optimal partial convergence. As shown later in this part, the parameter can be determined with the feedback intensity.

Substituting Eq. (3.35) into Eq. (3.19) yields the following expression for $\theta\left(\omega_p\right)$:

$$\theta\left(\omega_p\right) = \left[1 - \beta\left(\omega_p\right) - \gamma - \beta\left(\omega_p\right)\frac{3\gamma}{\omega_p^2}\right] f_{TS}\left(\omega_p\right) + \beta\left(\omega_p\right) f_{NDA}\left(\omega_p\right)\,. \tag{3.36}$$

The iteration eigenvalue of the relatively flat Fourier mode is:

$$f_{TS}\left(\omega_p\right) \approx 1\,, \tag{3.37}$$

with $\omega_p \approx 0$. To minimize the magnitude of $\theta(\omega_p)$ in Eq. (3.36) is equivalent to setting the coefficient of f_{TS}:

$$1 - \beta\left(\omega_p\right) - \gamma - \beta\left(\omega_p\right)\frac{3\gamma}{\omega_p^2} = 0\,. \tag{3.38}$$

Rearranging this equation yields:

$$\beta\left(\omega_p\right) = \frac{1-\gamma}{1+\frac{3\gamma}{\omega_p^2}}\,. \tag{3.39}$$

Since

$$\beta\left(\omega_p\right) = 1 - \Lambda^L(\omega_p)\,, \tag{3.35 revisited}$$

$$\Lambda(\omega_p) = \frac{1-c'}{1-c'+\frac{\omega_p^2}{3}} \, , \tag{3.24a revisited}$$

$$c' = 1-(1-c)(1-r) \, , \tag{3.20c revisited}$$

the nearly-optimal WS ratio r and the power iteration number L to achieve the nearly-optimal partial convergence can then be calculated by:

$$r = 1 - \frac{\left[1-\beta\left(\omega_p\right)\right]^{\frac{1}{L}}}{1-\left[1-\beta\left(\omega_p\right)\right]^{\frac{1}{L}}} \frac{\omega_p^2}{3(1-c)} \, , \tag{3.40a}$$

or

$$L = Log_{\Lambda(\omega_p)}\left[1-\beta\left(\omega_p\right)\right] \, . \tag{3.40b}$$

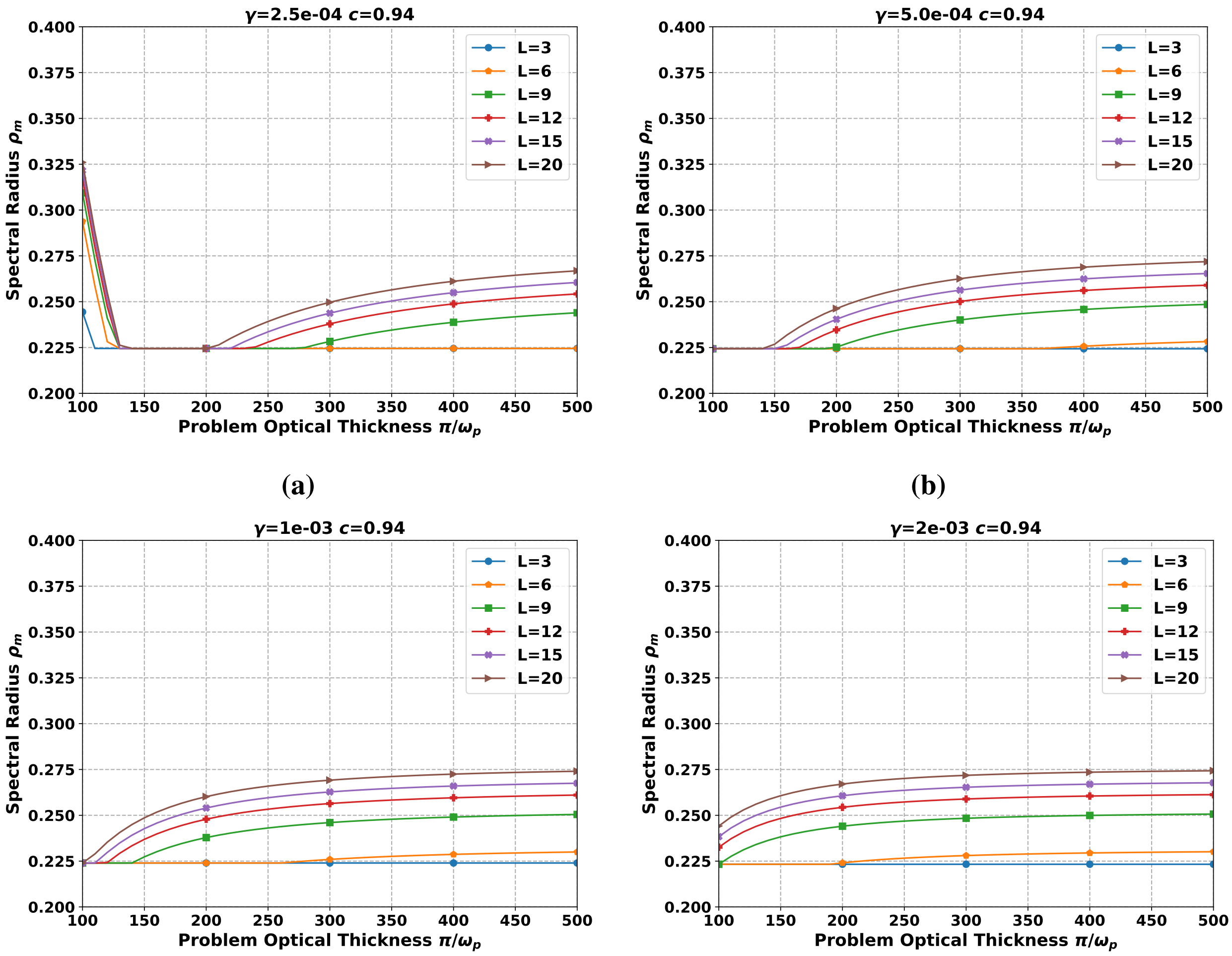
The formula to determine the optimal r is preferred to the formula for L because r can take any value in the range $[0,1)$, but L can only take discrete integer values. This means r has more flexibility to achieve the optimum.

The nearly-optimal partial convergence is problem-dependent. It depends on the size and the feedback intensity of the problem. Upon revisiting the results in Figures 3.2 and 3.3, it can be observed that there is a shared region for the partial convergence for problems of size $\Sigma_t X = 100$ and $\Sigma_t X = 300$–where the method has the same spectral radius as the classical NDA method. Therefore, it should be possible to find a problem-independent ω_p, or ω_p dependent only on the feedback, to estimate the nearly-optimal partial convergence irrespective of the problem domain size.

Two important reasons for searching for a ω_p independent of the problem size are that in practical problems, (i) it is difficult to define the equivalent model optical thickness; and (ii) even the partial convergence estimated using the optical thickness of the simulated problem may not be suitable since the practical problem is not fully consistent with the model problem (which is a homogeneous, mono-energetic, reflective problem with isotropic scattering). Consequently, our motivation to heuristically determine a limiting ω_p is analogous to an attempt to recover generality that is lost in the simplifications made to define the model problem. As we show in Chapter 4, this approach is successful.

To better illustrate the relationship between ω_p and the stability of the iteration scheme, the maximum spectral radius ρ_m of the partially converged NDA in problems with sizes ranging from $\Sigma_t X = 100$ to $\Sigma_t X = 500$ (with various ω_p and L used to estimate the partial convergence) are plotted and shown in Figures 3.5. The ρ_m is defined as:

$$\rho_m = \max_{\Sigma_t X \in [100,500]} \rho\left(\Sigma_t X, r(\omega_p)\right) \, . \tag{3.41}$$



(a)

(b)

(c)

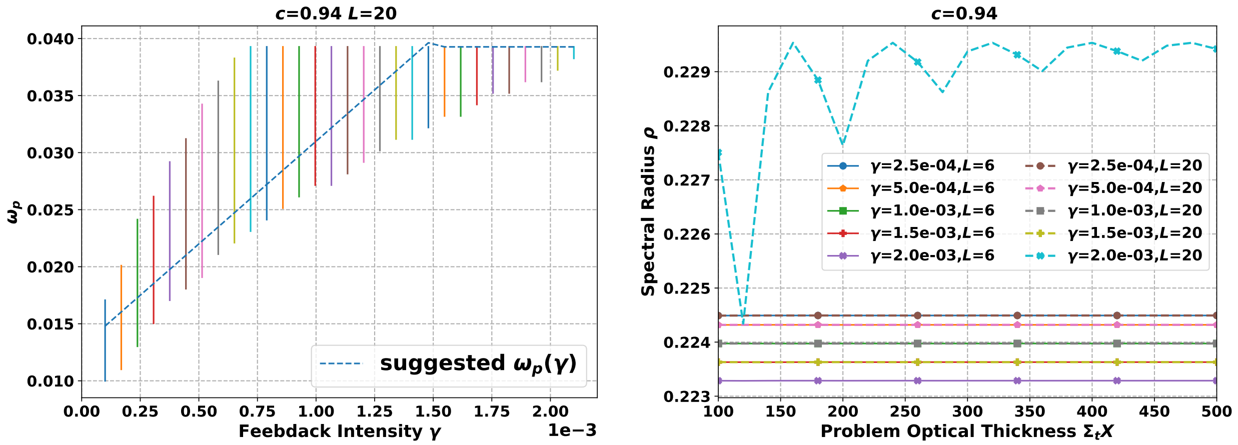
(d)

Figure 3.5: Relation between the maximum spectral radius and the ω_p used to estimate the nearly-optimal partial convergence. The feedback intensities γ of these four figures are $2.5 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}$ and 2×10^{-3}.

It can be seen that when the $L \leq 6$, ρ_m is close to the optimal threshold of 0.2247. However, there is no universal ω_p that can make ρ_m the same as 0.2247 for all cases. Nevertheless, the deviation of ρ_m from 0.2247 is relatively small.

When a fixed ω_p is desired for different problem sizes, it is suggested that $\omega_p \in [\frac{\pi}{200}, \frac{\pi}{150}]$ is a reasonable parameter range to estimate the nearly-optimal partial convergence. The ρ_m in this range is ~0.274. Compared to the classical NDA, the total iteration number to reach the convergence criteria ϵ for the partially converged NDA with $\rho = 0.274$ is increased by:

$$\eta = \frac{log_{0.274}\epsilon}{log_{0.2247}\epsilon} - 1 = \frac{ln(0.2247)}{ln(0.274)} - 1 \approx 15\% \,. \tag{3.42}$$



(a)

(b)

Figure 3.6: Investigation and verification of the ω_p to estimate the nearly-optimal partial convergence. (a) shows that the applicable range of ω_p for various feedback intensities. The vertical line representing the application range of ω_p for feedback intensity γ. The dashed line is the plot of the formula Eq. (3.43) to determine the ω_p in the applicable range based on γ. (b) shows that the spectral radius of the NDA with the partial convergence estimated by Algorithm 9 is nearly-optimal.

For the case with a typical convergence criteria of $\epsilon = 10^{-7}$, the total number of outer transport iterations, N, is increased by 1. Therefore, the overhead of a single additional iteration out of $O(10)$ may be considered negligible.

If a better nearly-optimal partial convergence estimation is desired, the ω_p can be determined based on the feedback intensity γ. The applicable range of ω_p with $L = 20$ that allows the partially converged NDA to achieve the spectral radius close to the classical NDA is searched and is illustrated in Figure 3.6a. The formula

$$\omega_p(\gamma) = \begin{cases} 18\gamma + 0.013\,, & \gamma \in [10^{-4}, 1.5 \times 10^{-3}]\,, \\ \pi/80\,, & \gamma > 1.5 \times 10^{-3}\,, \end{cases} \tag{3.43}$$

is suggested to be used from this study. The plot of this expression is the dashed line in Figures 3.6.

The algorithm to determine the nearly-optimal partial convergence is now presented and shown in Algorithm 9. The estimate r should not be very close to 1 because the linear system, Eq. (Alg 4.1a), will become nearly singular and difficult to solve by most conventional iterative methods. In this study, the upper bound of the WS ratio r is set to be 0.98 [6]. The WS ratio cannot be smaller than 0, either. Otherwise, the eigenvalue from the power iteration may not converge to the fundamental mode. Therefore, when the calculated r is too small i.e., in this case $r < 0.1$, the WS ratio will be set to 0.1 and the maximum number of power iterations will be updated by Eq. (3.40b). The minimum power iteration number L_m is also introduced as a lower limit.

Algorithm 9 Nearly-optimal partial convergence estimation.

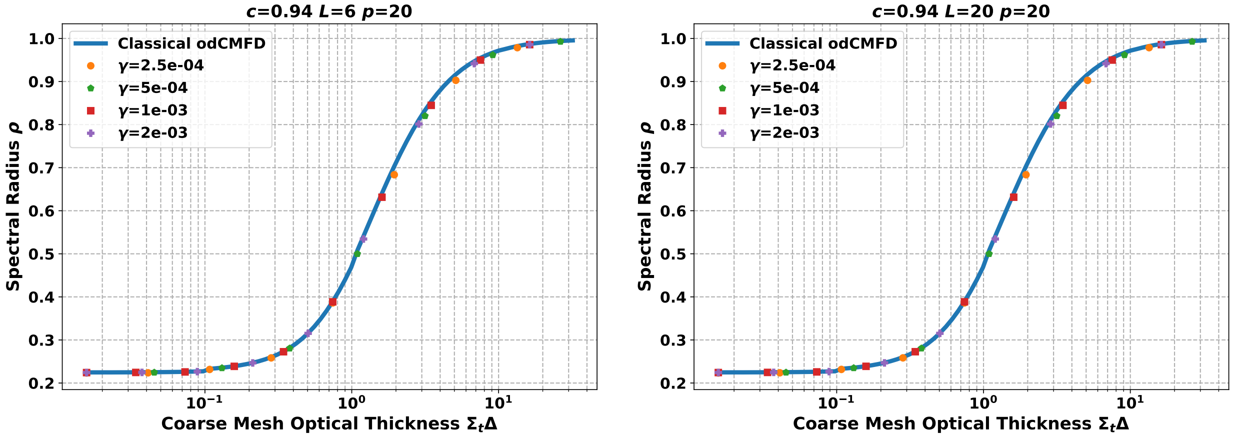
```
Input:PI Number $L$; minimum PI number $L_m$; feedback intensity $\gamma$.
Output:WS ratio $r$, PI Number $L$.
Calculate $\omega_p$ with Eq. (3.43), or use a fixed $\omega_p$ from the suggested range $[\frac{\pi}{200}, \frac{\pi}{150}]$.
Calculate the WS ratio $r$ with Eq. (3.40a), and make the $r$ in a rational range by:
    $r \leftarrow \min(r, 0.98)\,.$
if $r < 0.1$ then
    $r \leftarrow 0.1$.
    Update the power iteration number $L$ using Eq. (3.40b), and
        $L = \max(L, L_m)\,.$
end if
```

3.3.2 Stability of NOPC-stabilized Method

The spectral radius of NDA with nearly-optimal partial convergence, estimated using the feedback-dependent ω_p in Eq. (3.43), is shown in Figure 3.6b. It is observed that in all the cases, the iteration scheme has a very similar or even smaller spectral radius compared with the classical NDA. Therefore, the new relaxation-free NDA method for problems with feedback in which the low-order diffusion problem is nearly-optimally partially converged is named as nearly-optimal partially converged NDA.

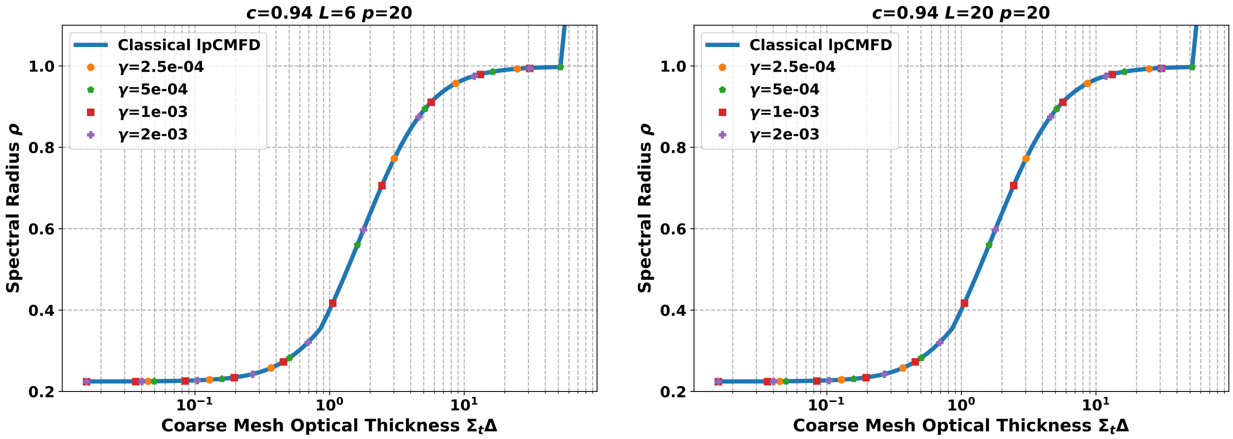
The algorithm is next applied to the discretized model problem. And the method is referred to as the NOPC-CMFD. We first apply this to the conventional CMFD method, where the flat prolongation is used for updating the transport solutions. The procedures developed in [16] are used here to optimize the diffusion coefficients. The spectral radius results are shown in Figures 3.7. From these results, it is observed that in utilizing the nearly-optimal partial convergence, the optimally diffusive Optimally Diffusive CMFD (odCMFD) problem with feedback can have almost identical convergence behavior compared to odCMFD without feedback. Therefore, we conclude that the nearly-optimal partial convergence method can (theoretically) make the iteration scheme in the practical simulations with feedback have the same performance as the classical CMFD in the simulations without feedback. Again we demonstrate this numerically in Chapter 4.

Next, we apply Algorithm 9 to the lpCMFD [95, 18] where the linear prolongation is used to update the fine mesh transport solutions. The results are shown in Figures 3.8. Here, it is observed that not only does the lpCMFD with the nearly-optimal partial convergence have almost the same spectral radius as lpCMFD for the region where lpCMFD is stable, but that it is also more stable than lpCMFD without feedback for the problem where the coarse mesh size is quite large. The odCMFD or lpCMFD with nearly-optimal partial convergence is called nearly-optimal partially converged odCMFD or nearly-optimal partially converged lpCMFD. It can be concluded that the nearly-optimal partially converged lpCMFD is a potentially unconditionally stable variant of the CMFD method for problems with feedback.



(a) **(b)**

Figure 3.7: Stability of the nearly-optimally partially converged odCMFD. L in (a) and (b) are 6 and 20, respectively.



(a) **(b)**

Figure 3.8: Stability of the nearly-optimally partially converged lpCMFD. L in (a) and (b) are 6 and 20, respectively.

3.3.3 Discussions of the Relaxation-free Method

3.3.3.1 Applicable Range

In the previous sections, the stability of the new relaxation-free method based on the nearly-optimal partial convergence is investigated in a relatively small parameter space. In this section, the stability of the scheme over a wider range of feedback intensities, γ, and scattering ratios, c, for NDA are studied. The predefined number of power iterations L in Algorithm 9 is 20 and L_m is 5. The problem size is assumed to be as large as $\Sigma_t X = 500$, which is larger than the size of typical nuclear reactors. These results are shown in Figure 3.9. It can be seen in Figure 3.9 that the new algorithm is predicted to be stable for a wide range of conditions with different scattering ratios and orders of feedback intensity.

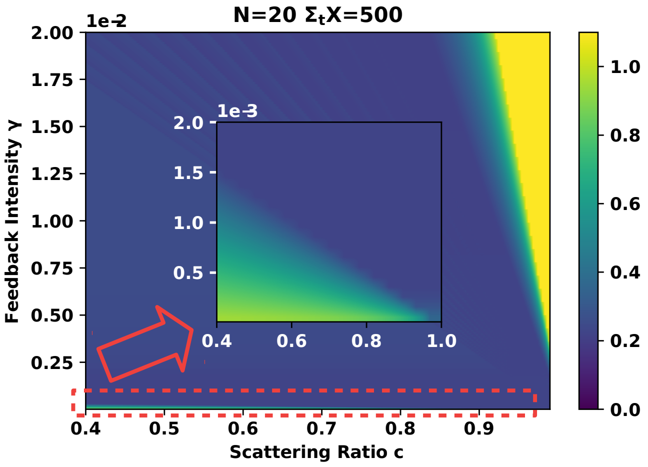


Figure 3.9: Spectral radius of NDA with Algorithm 9 in problems with various combinations of c and γ.

However, the method converges slowly or becomes unstable in the region with small c and γ, or with larger c and large γ. When the scattering ratio is smaller than 0.9 and feedback intensity is smaller than 1.5×10^{-3}, the spectral radius will not reach the optimal value of 0.2247, though it is smaller than 1. A more aggressive WS parameter can resolve this issue when r is allowed to be larger than 0.98. Nevertheless, in this parameter space, the method will not perform worse than methods using power iteration, that are also limited by the aggressiveness of the WS parameter. When γ is larger than 2.5×10^{-3}, the scheme will become unstable with a sufficiently large scattering ratio. Presently, it is not assumed that this is a big issue for the modeling of PWRs because the γ in a PWR seldom becomes larger than 1.5×10^{-3}, Furthermore, the scattering ratio of the fuel region is seldom smaller than 0.9. We would also expect the method to be suitable in fast reactor analysis where scattering ratios are smaller. However, for other highly scattering, and therefore thermal, reactor designs that might have extremely strong or sensitive reactivities, this numerical scheme will likely have reduced efficacy. Although, a problem with a large value for γ and high scattering ratio is likely not representative of a sensible nuclear reactor design. The definition of γ is

$$\gamma = \left(\frac{\Sigma_{a,1}}{\Sigma_{a,0}} - \frac{\Sigma_{f,1}}{\Sigma_{f,0}}\right)\frac{\Sigma_{a,0}}{\Sigma_{t,0}}\Phi_0 = \left(\frac{\Sigma_{a,1}}{\Sigma_{a,0}} - \frac{\Sigma_{f,1}}{\Sigma_{f,0}}\right)\Phi_0(1-c)\,. \tag{3.44}$$

This equation shows that as c increases, γ becomes smaller, so a design with large c and large γ is not likely possible.

3.3.3.2 Validity of Assumptions

In the model problem, it is assumed that γ is a constant across the whole problem domain. However, in realistic problems, the temperature distribution and the flux distribution are not flat. Since γ can be expressed as a local term dependent on the local flux and cross sections (or temperature). We devote substantial focus on how to define a suitable global γ to calculate the nearly-optimal partial convergence parameters in Chapter 4.

Another implicit assumption is that convergence to the fundamental mode eigenvalue is quite fast. However, there is a spectrum of eigenvalues in all problems, and a less aggressive WS parameter may affect the convergence to the dominant eigenvalue.

3.3.3.3 Implementation of the Relaxation-free Method

The relaxation-free CMFD method, i.e., nearly-optimal partially converged CMFD, should be easy to implement in high-fidelity whole-core simulation codes that already have something like Algorithm 8. The only modification compared to the current Picard scheme, is that some routines must be added to estimate the feedback intensity and estimate the nearly-optimal partial convergence. The overall structure of the iteration scheme is not altered. However, estimation of γ can be nontrivial as discussed in Chapter 4. Ultimately, we present this method as a relatively simple way to enhance the robustness of high-fidelity whole-core simulators, such as MPACT, for problems with feedback in the cross sections. Other methods are being developed and demonstrated to address robustness in multiphysics iterations [34, 38], but we consider these approaches to be more complex to adopt (sometimes considerably), and so far have only been shown to yield similar efficacy as to the method we propose here. Nevertheless, these other works also present important current future alternatives to the method described in this chapter. Another robust relaxation-free method in this thesis, X-CMFD, is investigated in Chapter 5.

3.3.3.4 Importance of the Relaxation-free Method

The Fourier analysis results show that the nearly-optimal partially converged NDA/CMFD method (or relaxation-free method), has a spectral radius comparable with the classical NDA/CMFD in a wide range of problems with feedback. The work is important because it enables the same effectiveness of NDA/CMFD acceleration for problems without feedback to that for problems with feedback. It eliminates the need for a user-defined and problem-dependent relaxation factor–where one does not know the optimal relaxation factor *a priori*. Additionally, from the neutronics solver perspective, the computational resource requirements will not vary much for the problems with and without feedback, so long as the solvers for the other physics are much less computationally expensive to evaluate compared to the neutronics solver. Lastly, the proposed method is developed from well-grounded theory and mathematical analysis techniques; it is not determined strictly through comparing results of numerical experiments and parametric studies.

3.4 Summary

In this chapter, the stability of partially converged NDA and CMFD to accelerate k-eigenvalue neutron transport problems with feedback was studied. The relationship of partial convergence and relaxation was derived, and a new relaxation-free iteration scheme was proposed. The results of the Fourier analysis are well-grounded and provide a theoretical foundation for the development of a robust relaxation-free multiphysics iteration scheme with NDA.

The investigation of the partial convergence of NDA and CMFD is motivated by the fact that the low-order diffusion eigenvalue problem in most applications of NDA and CMFD is typically solved with a limited number of WS power iterations. Therefore, the partial convergence is first theoretically parameterized with the number of power iterations and the WS ratio. The Fourier analysis results show that fully converging the NDA and CMFD is desired for problems without feedback, however, this approach is detrimental for problems with feedback. The theoretical results support the observation that the typical Picard iteration presented in the chapter is not unconditionally stable, and the convergence criteria on the low-order problem should be loosened to achieve good convergence rates.

The partial convergence is then related to a more conventional flux relaxation factor. It is shown that the partial convergence of the low-order problem introduces an implicit relaxation on the Fourier modes–an intuitive result of the partial convergence. The relaxation depends on the frequency of the Fourier modes and mainly affects the most slowly converging modes. This result is significant for three reasons. First, this work provides the theoretical explanation as to why partially converging the low-order problem of NDA/CMFD can stabilize the multiphysics iteration.

Second, it shows exactly how partially converging the low-order problem is inherently different from a constant relaxation factor. Finally, it predicts that partially converging NDA in problems with feedback can be done so in a way to achieve the same stability as NDA in problems without feedback. This result emphasizes the conclusion that partially converging the low-order problem rather than using relaxation should be the procedure adopted for the multiphysics simulation.

In light of these analysis results, the relaxation-free iteration scheme is proposed by deriving a formula to compute the WS ratio and the number of power iterations to use when solving the low-order diffusion at each outer iteration so that the overall spectral radius is practically minimized. It is also observed that the scheme could be further optimized through the procedure in [16] for CMFD with flat prolongation, or implemented directly without optimization for CMFD with linear prolongation. The relaxation-free method is also shown to have predicted convergence behavior similar to the NDA/CMFD acceleration in problems without feedback for a large range of problem parameters.

CHAPTER 4

Numerical Verification of Nearly-optimally Partially Converged CMFD

4.1 Introduction

In this chapter, we present the results of numerical studies based on the theoretical work in Chapter 3. The behavior of relaxation and partial convergence of diffusion solutions are first presented, and the properties of the feedback intensity in Chapter 3 are then investigated. The results indicate the importance of the partial convergence to stabilize the iteration scheme and verify the assumptions and predictions made. Next, implementation of the relaxation-free method in the MPACT [93], and numerical results of the method in problems of different scales with feedback of various forms are introduced. The new method is implemented based on the current CMFD solvers in MPACT. Compared to the original CMFD methods, the NOPC-CMFD method is much more stable for problems with feedback, and the convergence performance is demonstrated to be insensitive to the presence of feedback. The work presented provides well-grounded numerical verification of Chapter 3 and demonstrates that the proposed method works well for practical applications.

4.2 Overview of Iteration Schemes

4.2.1 Feedback from Other Physics

The feedback from three different physics: TH, critical boron search (CB), and equilibrium Xenon (EX), are of interest in this thesis. Coupling neutronics and feedback from other physics is essential for operational, safety, and design analysis of nuclear reactor cores.

The TH problem can be solved by some simplified-TH model [89], a more advanced subchannel TH code like CTF [90], or with computational fluid dynamics (CFD) codes. The critical boron search is used to find the boron concentration that yields a k-eigenvalue of unity (or some other target k). The equilibrium Xenon calculation updates the concentration of Xenon-135 and Iodine-135 inside the fuel region according to the fission power. More details about these physics and their solution can be found in [65].

The state variables, i.e., TH field, boron concentration, and Xenon and Iodine concentrations, are categorized into two groups: local state variables and non-local state variables. This designation relates to their dependence on the fission power. The boron concentration is a non-local state variable because it is only dependent on the latest estimate of the global k-eigenvalue in the calculation, and the feedback is non-local. The concentration of Xe-135, I-135, and the TH fields are local variables, and their feedback is local. It should be noted that the TH is also non-local because of convection, but it is only weakly non-local. The fuel temperatures behave much more like local feedback and typically dominate feedback effects in PWRs.

4.2.2 Wielandt Shifted Power Iteration in Practice

The CMFD equation, i.e. the equation of the low-order diffusion eigenvalue problem has the form as:

$$\mathbf{M}\boldsymbol{\phi} = \lambda \mathbf{F}\boldsymbol{\phi} \,. \tag{2.45 revisited}$$

Algorithm 4 shows how the CMFD equation is solved with WS power iteration. In practice, to balance the efficiency and convergence, the algorithm to perform the WS power iteration in MPACT, DeCART [2], nTRACER [3], STREAM [41], and possibly other CMFD-accelerated transport codes terminates the power iterations if l reaches the maximum number of power iterations L or if some other convergence criteria of the CMFD solution are satisfied.

Several parameters have been used to measure the convergence of the CMFD solutions. The first one is the relative residual f_{res} that is defined by:

$$f^l_{res} = \frac{\left\|(\mathbf{M} - \lambda^l_s\mathbf{F})\mathbf{\Phi} - (\lambda^{l-1} - \lambda^l_s)\mathbf{F}\mathbf{\Phi}^{l-1}\right\| / \left\|\mathbf{F}\mathbf{\Phi}^l\right\|}{\max\Big(\|(\mathbf{M} - \lambda^0_s\mathbf{F})\mathbf{\Phi}^0 - (\lambda^0 - \lambda^0_s)\mathbf{F}\mathbf{\Phi}^0\| / \|\mathbf{F}\mathbf{\Phi}^0\| \,, f(\epsilon_f)\Big)} \,, \tag{4.1}$$

in which $f(\epsilon_f)$ is a value determined by the fission source residual convergence specification ϵ_f, that is used for determining the global convergence and terminating the outer iterations. Other criteria are based on differences in the fission source and eigenvalue between two successive power iterations. The fission source difference and the eigenvalue difference are defined in Eqs. (4.2) and in Eq. (4.3), respectively.

$$\Delta f^l = \frac{\left\|\mathbf{F}\mathbf{\Phi}^l - \mathbf{F}\mathbf{\Phi}^{l-1}\right\|}{\|\mathbf{F}\mathbf{\Phi}^{l-1}\|} \,, \tag{4.2}$$

$$\Delta k^l = \left| \frac{1}{\lambda^l} - \frac{1}{\lambda^{l-1}} \right| . \tag{4.3}$$

The convergence criteria for f_{res}, Δf and Δk are represented by r_{tol}, f_{tol} and k_{tol}, respectively. In MPACT and DeCART, f_{res} and Δk are used. In STREAM, Δf and Δk are used with f_{tol} and k_{tol} being dynamically controlled depending on the convergence of the transport solver.

4.2.3 CMFD Solvers in MPACT

Two CMFD solvers based on the power iteration are implemented in MPACT and studied here. They are Multigroup CMFD solver with power iteration (MGPI) and MSED solver [32].

The default CMFD solver MGPI in MPACT solves the full space and energy multigroup diffusion problem. The linear system Eq. (Alg 4.1a) is solved using the GMRES [96, 74] solver from PETSc [97] preconditioned by the block Jacobi method. The solution process is dramatically slow for large-scale problems. For most applications with MPACT, the values for the maximum number of power iterations L, λ_s, r_{tol} and k_{tol} are 20, $2/3$, 10^{-2}, and 10^{-6}. These values were determined from parametric studies on representative problems to approximately optimize the trade-off between time spent in the CMFD problem and the convergence rate of the overall iteration scheme. For some simulations, the iteration-dependent λ_s, as introduced in [17], are also used.

A more efficient multilevel CMFD solver, the MSED solver [32], has been implemented in MPACT more recently and has been introduced in Algorithm 5 in Chapter 2. The upper bound of the number of power iterations performed for grey eigenvalue problem $L = L_{MG}L_{GR}$ can be much larger than the L used for solving the MGPI system. L_{MG} indicates the number of iterations between the multigroup level and grey diffusion level, and L_{GR} is used to indicate the number of grey power iterations performed on the grey diffusion level. The iteration-dependent λ_s is also used. These features make the MSED solver much more efficient than the MGPI. The same convergence criteria are used for the MSED and MGPI solvers. More details about the MSED method and its performance can be found in [32]. The thought behind developing a more efficient CMFD solver is that the overall run time of the iteration can be reduced significantly with a more efficient CMFD solver since the CMFD run time dominates the total run time (at least in MPACT). The thought for pursuing more accurate CMFD solutions is that it was believed the more tightly converged the CMFD solution is, the fewer outer iterations are needed to achieve the global convergence. However, as shown in Chapter 3, this latter assumption only holds when no feedback or weak feedback is present.

4.3 Nearly-optimally Partially Converged CMFD Solver

4.3.1 Importance of the Partial Convergence of CMFD solutions

Partially converging the CMFD solution to stabilize the iteration scheme is shown to be a better approach than introducing relaxation from the Fourier stability analyses in [39] and Chapter 3. This claim is further supported by the numerical results from simulating the VERA core physics benchmark progression problem 6 (P6) [98] with MPACT presented in this section. The P6 is a 3D PWR assembly problem that has been used by different codes to assess the performance and practicality of their multiphysics iteration schemes [84, 23, 33].

Before showing the effect of the partial convergence and relaxation, the number of outer iterations that MPACT takes to converge P6 with zero power and fully converging the CMFD solution is established. This value is **12**, and is considered to be the performance of MPACT with classical CMFD for P6. To assess the behavior of the partial convergence several P6 cases are used with varying total power. As shown later in this subsection, if the partial convergence of the CMFD solution is nearly-optimal, then MPACT can converge the P6 cases with feedback in 12 outer iterations.

4.3.1.1 Effect of Partial Convergence

Figure 4.1 illustrates how the partial convergence of the CMFD solution affects the convergence behavior of the coupled iteration scheme for P6. The P6 cases with 20% and 100% rated power are simulated with a fixed WS ratio of $r = 0.9$ and a varying number of power iterations L. The WS ratio is defined as:

$$r = \frac{\lambda_s}{\lambda}\,. \tag{3.12 revisited}$$

In Chapter 3, we have shown that the partial convergence of the CMFD solutions are determined by the WS ratio and the number of power iterations. No power relaxation is applied, and the simplified TH model [89] is used for feedback.

It can be seen that to converge the P6 cases in the minimum number of outer iterations, **12**, the partial convergence of the CMFD solutions can neither be too tight nor too loose. Moreover, there is an applicable range of the partial convergence of CMFD solution that allows MPACT to converge optimally. The applicable range is wider for the case with 20% rated power compared to the case with 100% rated power. These observations support the theoretical findings in Chapter 3– that there is an optimal range for the partial convergence of CMFD in problems with feedback that has practically the same convergence rate as classical CMFD in the same problem without feedback.

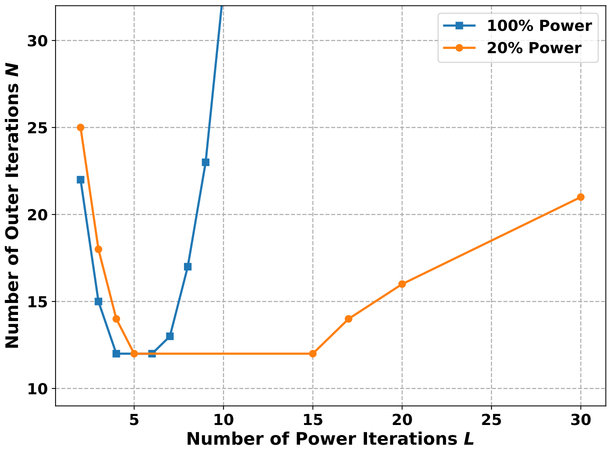


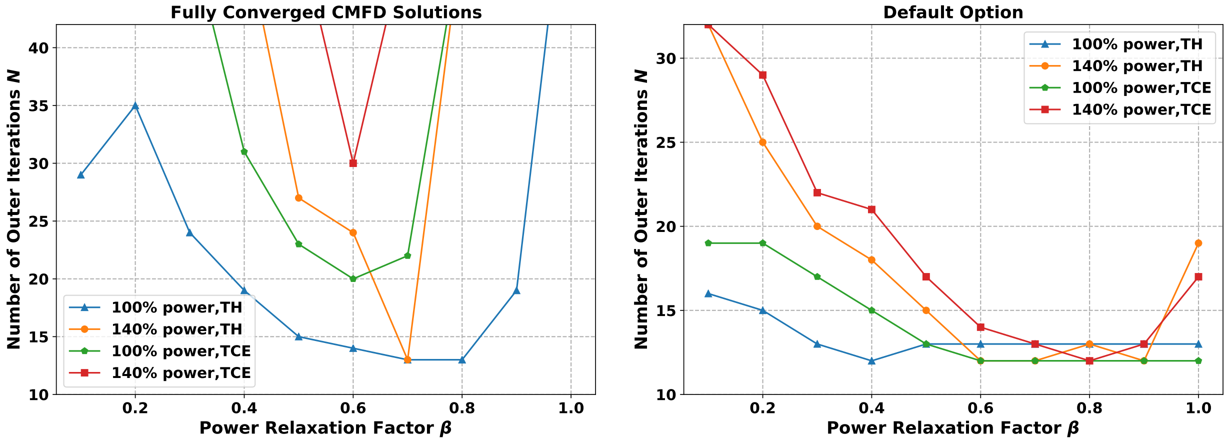
Figure 4.1: Relation between the number of total outer iterations to converge P6 and the number of WS power iterations. 12 is the minimum number of outer iterations MPACT requires to converge this problem. The feedback is from TH.

It can also be observed that when the CMFD equations are solved with 4 power iterations, then MPACT converges the case with 100% nominal power faster than the case with 20% nominal power. The observation supports the theoretical finding in Chapter 3 that a problem with stronger cross section feedback may converge faster than the problem with weaker, or no feedback–if partial convergence of the CMFD solution is relatively loose.

4.3.1.2 Effect of Relaxation

Though the effect of relaxation on the convergence rate of the Picard iteration scheme has been examined in several different works [20, 39, 31], none of these works have taken the partial convergence of CMFD solution into account. However, it should be noted that the Monte Carlo community frequently publishes work involving numerical experiments to assess the number of batches to run between cross section updates [84, 26, 99, 28]. This is notionally the same idea as the partial convergence of the fission source–although it differs with respect to the use of a diffusion acceleration scheme. In this subsection, the convergence behavior of MPACT in simulating the P6 cases using different relaxation factors with both fully converged and partially converged CMFD solutions is investigated. Here we focus on the cases with nominal power and 140% rated power, since it is not uncommon to have assembly peaking factors of 1.4 in full-core calculations. The results are illustrated in Figures 4.2.

When the CMFD solutions are fully converged for each P6 case, as illustrated in Figure 4.2a, there is an optimal relaxation factor where the total number of outer iterations is the smallest. This optimal factor varies with the intensity of feedback (i.e. the rated power). For the P6 problem with 100% nominal power and feedback from TH, the optimal relaxation factor is 0.8 (remarkably



(a) (b)

Figure 4.2: Convergence behavior of MPACT with varying relaxation factor. (a) shows the results for fully converged CMFD solutions. (b) shows the results with default CMFD that are partially converged.

close to historical values determined from parametric studies), while for the P6 problem with 140 % nominal power and feedback from TH, Critical boron search and Equilibrium xenon calculation (TCE), the optimal relaxation is 0.6. Moreover, the stability of the iteration scheme is highly sensitive to the relaxation factor. If the relaxation factor is varied by 0.1, the number of outer iterations to converge the P6 cases can change by more than 10 iterations when a nominal power of 140% is used. It can also be observed that the minimum number of outer iterations to converge the P6 case with 100% power is 20 when the CMFD is fully converged with feedback from TCE. This number of outer iterations is much larger than the no feedback value, which is 12. Historically, this may not have been of much concern in full-core nodal diffusion calculations since the cost of an iteration is fractions of a second. However, in the application of high-fidelity methods, the computational cost of a single iteration is non-trivial.

When the CMFD equations are partially solved, as illustrated in Figure 4.2b, not only the applicable range of the relaxation factor that makes the simulation stable is broader, but the minimal value of the total outer iteration numbers to converge is similar to that for the classical CMFD case with no feedback. Moreover, the convergence performance of the iteration scheme becomes less sensitive to the relaxation factor.

Therefore, the partial convergence of CMFD solutions is more important than the relaxation to stabilize the Picard iteration scheme in CMFD-accelerated transport calculations. There also exists a range of partial convergence that is nearly-optimal in the sense that the coupled problem may be converged in the same number of iterations as the uncoupled problem. In the following subsection, we investigate how to automatically determine a problem-dependent nearly-optimal partial convergence (NOPC) parameter, so that the iteration scheme is robust for a wide range of problems.

4.3.2 Estimation of the Nearly-optimal Partial Convergence

The primary issue that remains is how to estimate the degree of partial convergence of the CMFD solutions in a problem-dependent way. The theoretical results from Chapter 3 show that the NOPC is expressible in terms of the WS ratio r and the number of CMFD power iterations L defined by Eq. (3.40a).

$$r = 1 - \frac{(1-\beta)^{\frac{1}{L}}}{1-(1-\beta)^{\frac{1}{L}}} \frac{\omega_p^2}{3(1-c)} , \tag{3.40a revisited}$$

where

$$\beta = \frac{1-\gamma_g}{1+\frac{3\gamma}{\omega_p^2}} . \tag{3.39 revisited}$$

However, these expressions depend on knowing the problem dependent feedback intensity parameter, γ. In this subsection, we describe an approach to efficiently determine this value. One may use Eq. (3.40a) to determine an optimal r or L by fixing one and determining the other. Here we have chosen to fix L as it is a discrete integer value and determine r to achieve NOPC. By choosing to adjust the real number r over the discrete integer L, it should allow one to get closer to the optimal convergence rate. The NOPC-CMFD is applicable because λ_s is a natural part of solving the CMFD equations. In Eq. (3.40a), the most important parameter is γ_g, which is introduced to represent the global core-wise feedback intensity [39], and is called the global feedback intensity in this thesis. γ_g is problem-dependent and, for practical reasons, is determined by numerical estimation. c is the scattering ratio. The ω_p is a problem-dependent parameter characterizing the total optical thickness of the modeled problem (not the optical thickness of a spatially discretized cell). As shown in Chapter 3, a fixed ω_p or a γ_g-dependent ω_p can be used with little degradation of performance.

The γ_g used in Eq. (3.40a) must be determined from the localized feedback intensity γ. In Chapter 3, the expression for γ is

$$\gamma = \left(\frac{d\Sigma_a}{d\Phi}\frac{1}{\Sigma_a} - \frac{d\Sigma_f}{d\Phi}\frac{1}{\Sigma_f} \right) \Phi(1-c) . \tag{3.16h revisited}$$

This formula is derived from the NDA method for a homogeneous problem, where the low-order equation is solved on the same spatial grid as the transport problem–rather than on a coarser homogenized spatial grid in a heterogeneous problem. Therefore, in [39] and Chapter 3, γ_g is γ. This equality **does not** hold in the realistic problem because γ is space-dependent, since the derivative terms are space-dependent. The process to calculate the localized feedback intensity γ and the process to determine the γ_g based on γ are described later in this section.

In the Fourier analysis, the NOPC determined by the formula of Eq. (3.40a) results in practically the same convergence rate in the feedback problem as in the problem without feedback. However, in the derivation process of the NOPC formula [39], several properties of the feedback intensities γ and γ_g have been assumed.

1. **Locality**: The γ must be local. The word *locality* refers to the range inside one coarse mesh cell. It indicates that the physics determining the cross section feedback is mainly affected by the local neutron flux (or more specifically, the local fission rate).

2. **Non-negativity**: The γ_g of the problem is non-negative. A positive γ_g denotes the negative feedback of the reactor design, which is generally a requirement for reactor designs.

3. **Positive correlation**: The γ_g should be increasing when increasing the power at which the problem is simulated. Moreover, γ_g should be nearly proportional to the operating power of the reactor when the feedback is only from TH.

In the following subsection, we verify these properties of the estimated γ and γ_g numerically.

4.3.3 Estimation of the Localized Feedback Intensity

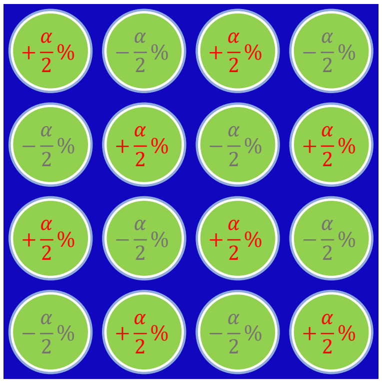


Figure 4.3: Illustration of the Red-Black perturbation.

As shown in Eq. (3.16h), γ is a term related to the derivative of the cross section with respect to the local scalar flux. A natural way to approximate these derivatives is to use the finite difference method with local perturbations. The algorithm to perturb the scalar flux to estimate the cross section derivatives for γ is given by Algorithm 10.

The perturbation is applied in a **Red-Black** fashion as illustrated in Figure 4.3, so that the γ of all the coarse meshes can be calculated simultaneously. The details are shown in Algorithm 10,

Algorithm 10 Perturbing the flux in MPACT for calculating feedback intensity.

```
Input lPerturb {Flag to indicate whether the flux is perturbed}
for $j_1 = 1, 2, \cdots$ , nPin do
    Calculate the perturbation factor:
        $$\eta = 1 + \Big[\mathbf{mod}(j_1, 2) - 0.5)\Big] \times \alpha\% \,. \qquad \text{(Alg 10.1)}$$
    for $C_1 = 1, 2, \cdots$ , nCoarseMesh(j$_1$) do
        for $f_1 = 1, 2, \cdots$ , nFineMesh(C$_1$) do
            for $g = 1, 2, \cdots G$ do
                if lPerturb == True then
                    $\phi_{g,f_1} = \eta\phi_{g,f_1}$ ,
                else
                    $\phi_{g,f_1} = \frac{\phi_{g,f_1}}{\eta}$ .
                end if
            end for
        end for
    end for
end for
```

where nPin is the number of pin-cells in the model; nCoarseMesh($\mathtt{j_1}$) is the number of the coarse meshes inside the j_1^{th} pin-cell; nFineMesh($\mathtt{C_1}$) is the number of the fine meshes inside the C_1^{th} coarse mesh. g is the index of the energy group with G as the total number of energy groups. As indicated by Eq. (Alg 10.1), the fine mesh fluxes of the pin-cell with the even index are perturbed by a factor $\frac{\alpha}{2}\%$, and those of the pin-cell with the odd index are perturbed by $-\frac{\alpha}{2}\%$. α is a small value to make the Eq. (4.5) hold and make the computational effort less when calculating the other physics and applying feedback during the perturbation. The value of α used will be discussed later in this section. When the fine mesh flux is perturbed in the **Red-Black** fashion, the averaged TH parameters of their neighboring pin-cells will vary little. Thus, the perturbation of the feedback is induced by the variation of the localized flux. Additionally, the perturbed fine mesh flux can be reset to its unperturbed value without storing the perturbed flux.

The reason that the Red-Black perturbation is applied based on a pin-cell index rather than coarse mesh index is because the coarse mesh may be smaller than a pin-cell (e.g. a quarter pin-cell). In this situation the Red-Black sequence cancels out the perturbations over the pin-cell. As a result, the real γ estimated will not reflect the actual feedback intensity.

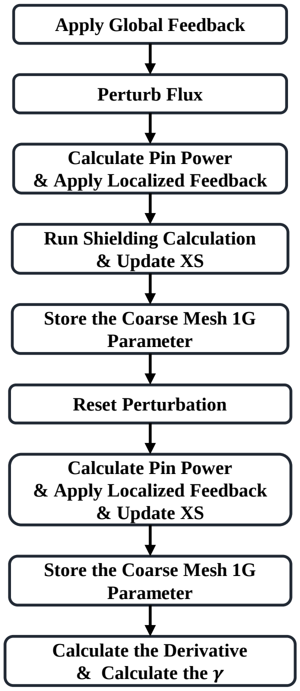


Figure 4.4: Flowchart of feedback intensity calculation. Shielding calculation is only performed once. But the solutions of the localized feedback needed to be performed twice.

The feedback intensity is calculated with the procedure shown in Figure 4.4. The derivatives of all the coarse mesh cells are approximated by:

$$\frac{d\Sigma_i}{d\Phi} \approx \frac{\Sigma_{ip} - \Sigma_{i0}}{\phi_{ip} - \phi_{i0}}\,, \tag{4.5}$$

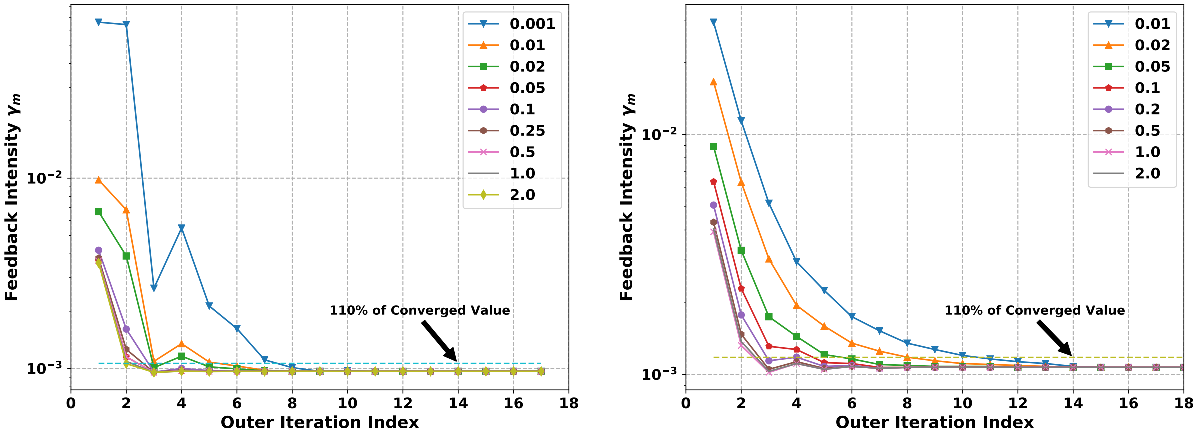
where the subscript p stands for perturbation, and the subscript 0 stands for the unperturbed state. It should be noted that this approach is akin to obtaining a numerical estimate of a part of the Jacobian of the multiphysics problem. It is also important to note that the procedure results in a straightforward estimate of the Jacobian that is ultimately represented by a scalar quantity after spatial integration.

Our next consideration in the application of Algorithm 10 is the value of α required so that a relatively accurate γ can be estimated once at the beginning of the calculation. Two problems from [98] are used to determine an appropriate range of values for α. One is a 3D pin-cell problem developed by extruding the 2D pin-cell model specified by the VERA core physics benchmark problem 1a (3D P1). The second problem is P6. Both of these problems are simulated with a peaking factor of 1.4. Only the feedback from TH is assumed to be present. The TH feedback is again provided by the simplified TH model.

During the simulations, the feedback intensity is estimated with different values of α for every outer iteration. The maximum value γ_m and the value weighted by the fission rate $\bar{\gamma}$, calculated by Eq. (4.6), are selected to characterize the global feedback intensity of the problem.

$$\bar{\gamma} = \frac{\sum_c V_c \nu \Sigma_{fc} \Phi_c \gamma(c, \Phi_c)}{\sum_c V_c \nu \Sigma_{fc} \Phi_c} \,. \tag{4.6}$$

We refer to each of these quantities as representing the *characteristic* feedback intensity of the model. As we will show later, these two variables are good approximations of the global feedback intensity, γ_g. Plots in Figures 4.5 show how the estimated γ_m varies with the perturbation factor.



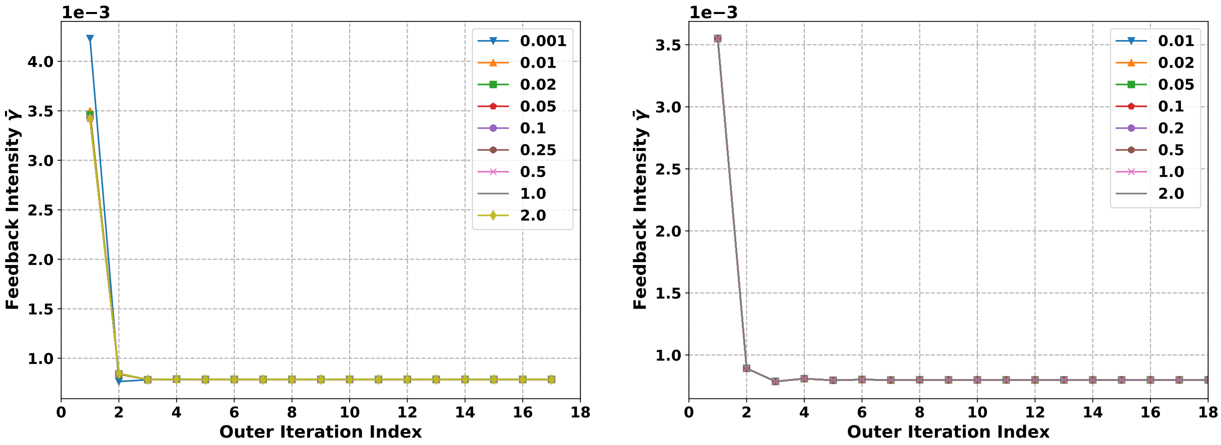
(a) **(b)**

Figure 4.5: The variation of the estimated γ_m using different perturbation factors α during the iteration. (a) shows the results for 3D P1 problem. (b) shows the results for P6 problem.

For both problems, the trend of the variation of γ_m as a function of iteration number is similar. It can also be seen that for both cases, all the estimated γ_m eventually converge to the same value at the 16^{th} outer iteration. However, the rate at which the estimated γ_m approaches the converged value is affected by the magnitude of α. When $\alpha < 0.2$, the estimated γ_m lies outside 10% of the converged value at the 4^{th} outer iteration. The smaller α is, the slower the rate that the iterative γ_m converges to the final value. The result indicates that α cannot be too small, or it cannot be accurately distinguished from the iterative error in the calculation early on in the solution process. Therefore, the $\alpha = 0.5$ is used to get a relatively accurate γ_m before the 3^{rd} iteration.

Plots in Figures 4.6 illustrate how the estimated $\bar{\gamma}$ varies with the perturbation factor, α. Compared to γ_m, $\bar{\gamma}$ approaches the converged value quite a bit faster in both problems. The curves of the weighted feedback intensity estimated with different perturbation factors overlap each other, indicating the effect of the magnitude of α on $\bar{\gamma}$ is practically negligible.

As will be shown later in Section 4.3.5.2, γ_m and $\bar{\gamma}$ may be used to represent γ_g.



(a) **(b)**

Figure 4.6: The variation of the estimated $\bar{\gamma}$ using different perturbation factors α during the iteration. (a) shows the results for 3D P1 problem. (b) shows the results for P6 problem.

4.3.4 Verification of Feedback Intensity

It is important to check whether the estimated γ and γ_g satisfy the assumptions made in the Fourier analysis in [39] and Chapter 3. Furthermore, as mentioned in the previous subsection, we also wish to verify these assumptions for the realistic problems of interest. The locality of γ has been satisfied automatically during the calculation process. The non-negativity of γ_g is verified by the results in Figures 4.5 and Figures 4.6. The positive-correlation is then verified on the P6 cases in this part.

The P6 cases are simulated with feedback from TH and TCE. The power of the cases is varied from 0.1% to 140% of the rated power. The results are shown in Figure 4.7. It can be observed that in both problems, the estimated characteristic feedback intensity increases with power. Therefore, the estimated γ_g has a positive correlation with respect to power.

When the problems are simulated with feedback from TH, the relation between the feedback intensity and the power is well represented by a linear fit. The expression for the fit and R^2 are illustrated together in Figure 4.7. The R^2 is very close to 1, indicating that for the modeled problem, the feedback intensity is nearly linearly dependent on the power simulated. This agrees with what has been found previously for a pin-cell problem in [39],

When the power is very small, the feedback intensity is close to 0. This observation agrees with the intuitive physics of the reactor–that the cross sections of the pin-cells are almost constant near zero-power. The differences in the estimated γ_m or γ_g from TCE and TH are increasing from 0 to a maximum value and then decreasing with increasing power. When power is larger than 20% of the nominal power, the differences are significant. We assert that it is for this reason that the

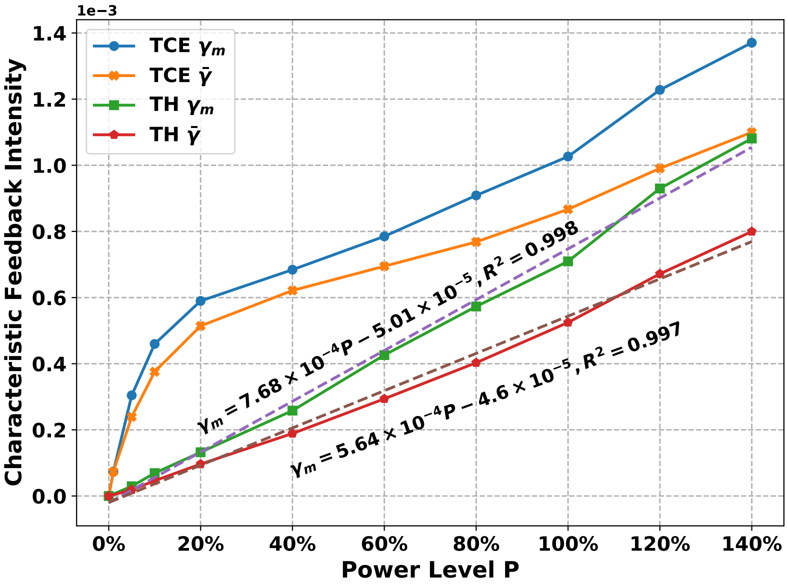


Figure 4.7: The relation between the characteristic feedback intensity and the power for different feedbacks.

problem modeled becomes harder to converge when the CMFD solutions are fully converged as shown in Figure 4.2a, if the feedback from equilibrium xenon is also taken into account.

The variation of the differences in the characteristic feedback intensity with respect to power for cases with feedback from TCE and TH, and the nonlinear shape of the relation between the feedback intensity and the power when feedback is from TCE, can be explained by the equilibrium xenon model. The equilibrium xenon concentration is calculated by

$$N_{Xe} = \frac{(\lambda_I + \lambda_{Xe})\Sigma_f \Phi}{\lambda_{Xe} + \sigma_a^{Xe}\Phi}\,, \tag{4.7}$$

where N_{Xe} is the number density of xenon atoms in a spatial cell, λ is the radioactive decay constant, and σ_{Xe} is the microscopic absorption cross section of xenon. Assuming that the presence of xenon only affects the absorption cross section, then the additional feedback intensity is

$$\Delta\gamma \propto \Phi \frac{\partial N_{Xe}}{\partial \Phi} = \frac{\lambda_{Xe}(\lambda_I + \lambda_{Xe})\Sigma_f}{(\lambda_{Xe} + \sigma_a^{Xe}\Phi)^2}\Phi\,. \tag{4.8}$$

The $\Delta\gamma$ will increase from 0 to a maximum value and then decrease to zero with increasing power.

In summary, the work presented in this section not only reproduces the nearly-linear relation of the feedback intensity to the power when only the feedback from TH is present, but also presents the non-linearity of the γ_g-power relationship when feedback is from TCE. This non-linearity is explained with the equilibrium xenon model.

4.3.5 Flowchart of the NOPC-CMFD

4.3.5.1 Overall Procedure

The implementation details of the new relaxation-free scheme–the NOPC-CMFD method–are illustrated here. The overall flowchart is shown in Figure 4.8. Compared with the original iteration scheme shown in Figure 2.1, it can be found that the only modification is to add several modules to calculate the feedback intensity and determine the NOPC.

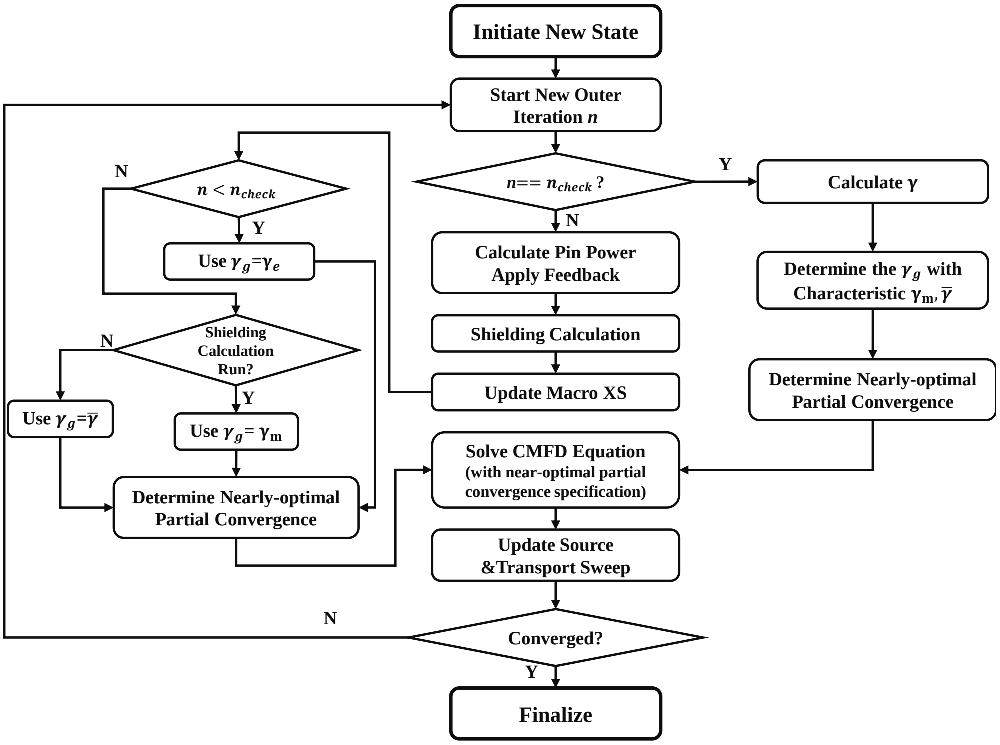

Initiate New State
Start New Outer Iteration n
n == n_{check} ?
Y
N
Calculate γ
Determine the γ_g with Characteristic γ_m, $\bar{\gamma}$
Determine Nearly-optimal Partial Convergence
N
$n < n_{check}$
Y
Use γ_g=γ_e
N
Shielding Calculation Run?
Y
Use γ_g=$\bar{\gamma}$
Use γ_g= γ_m
Determine Nearly-optimal Partial Convergence
Calculate Pin Power Apply Feedback
Shielding Calculation
Update Macro XS
Solve CMFD Equation (with near-optimal partial convergence specification)
Update Source &Transport Sweep
N
Converged?
Y
Finalize

Figure 4.8: Flowchart of NOPC-CMFD iteration scheme. “XS” stands for cross section. For brevity, the procedures for “Transfer Power”, “Solve other Physics” and “Homogenize” are not shown in the flowchart, compared to Figure 2.1.

In the new scheme, MPACT estimates the feedback intensity at the beginning of the iteration with index n_{check}. As investigated in Figures 4.6, the suggested value of n_{check} is 4. For the multi-state calculation, the n_{check} can be set to 2 for the states other than the first state. Before the γ and γ_g are estimated, the default γ_e is used, which may be specified by a user or code default. As suggested in Figures 4.6, the feedback intensity for the first few states is very large because the cross sections and state variables are far from their converged values. To accommodate this–and to enhance stability–a default feedback intensity should be chosen that bounds the maximum estimated feedback intensity. Based on the results in Figure 4.7, setting γ_e to be 1.5×10^{-3} is a reasonable choice for PWR analysis.

After the feedback intensity is estimated, the WS ratio that provides the NOPC (Eq. (3.40a)) can be calculated. Another practical consideration for the estimation of the feedback intensity that has so far been overlooked, is the effect of the resonance self-shielding calculation. If the shielding calculation is run at the beginning of the outer iteration, the γ_m is used to estimate the partial convergence, and the $\bar{\gamma}$ is used otherwise. This practice is supported by the sensitivity study results shown in Section 4.3.5.2.

This NOPC modification can be applied to any methods for solving the CMFD equations, so long as the Wielandt shifted power iteration is used to obtain the eigenvalue. To demonstrate this point, we provide numerical results for the default Multigroup CMFD solver with power iteration in MPACT, and also for the more advanced Multilevel in Space and Energy Diffusion (MSED) solver. In discussing the results in the following sections, we refer to the MGPI with the NOPC specification as MGPC (M̲ultigroup C̲MFD solver with the nearly-optimal p̲artial c̲onvergence specification), and MSED with NOPC specification as MSPC (MSED solver with the nearly-optimal p̲artial c̲onvergence specification). Of particular interest to the authors is also the performance of MSPC compared to MSEDL that was developed in [32] to enhance robustness of MSED in multiphysics problems. Note that the MGPI solver and MSED solver are different in the solution process. The basic CMFD equation is derived based on the odCMFD method [16].

4.3.5.2 Sensitivity Study of Feedback Intensity

The motivation to use γ_m and $\bar{\gamma}$ for estimating the NOPC is that these two parameters are the most natural representatives of the global pin-cell-wise feedback intensities. However, these two parameters may not be the best estimation of the global feedback intensity.

On one hand, γ_m represents the maximum localized feedback intensity over all pin-cells. If γ_m is used to estimate the NOPC for CMFD, it will ensure the scheme is stable. However, the partial convergence determined by γ_m may be too conservative to achieve the best overall convergence. On the other hand, the $\bar{\gamma}$ is a metric of the averaged feedback intensity. Therefore the estimated value is smaller than γ_m, and the partial convergence will be tighter. Using the partial convergence estimated by $\bar{\gamma}$ however, may still make the overall scheme unstable. These statements emphasize the importance of calculating γ_g so that one has a fast and robust iteration scheme, and further illustrates that the robustness of this scheme depends on the robustness and accuracy of estimating γ_g.

To gain a better understanding of how γ_g influences the overall performance and stability of convergence in practical problems, two sensitivity studies are performed by investigating the convergence behavior of the iteration scheme with γ_g varying in the ranges determined by these two characteristic parameters. The P6 cases with nominal powers of 100% and 140%, and with feedback from TH, TC, and TCE, are simulated with MSPC as the CMFD solver. In both studies, the value of γ_g used to estimate the partial convergence is assumed to be $\eta\gamma_m$ with η varying in the range $[0.8, 1.5]$. The number of total outer iterations to converge the problem is recorded.

For the first study, the resonance self-shielding calculation is performed at each outer iteration. The sensitivity study is based on the γ_m, because in the shielding calculation, the parameters (such as background cross section) to update the cross sections are updated with the TH condition of its neighboring cells. Although the feedback investigated in this research is local, the presence of the shielding calculation can propagate the effect of feedback through changes in the local spectrum or the Dancoff factors. As a result, the γ_m could be a better indicator of the γ_g used to estimate the NOPC. And if not, a correction factor can be applied.

The results are shown in Table 4.1, indicating that γ_m is a good estimate for the global feedback intensity. For most of the cases, across the range of γ_g investigated, the number of total outer iterations to converge the problem is the minimum with the partial convergence specification estimated with γ_m. The number of iteration in this case is the same as the optimal number of outer iterations to converge the P6 without feedback. When the power is 140% rated power and feedback is from TH or TCE, it takes fewer outer iterations if the partial convergence is estimated with a γ_g that is larger than γ_m. However, the overhead of the total outer iteration number compared to the optimal value is smaller than 2. Therefore, γ_m is a reasonable option to use for estimation of the NOPC parameters when the resonance self-shielding calculation is used to update the cross sections at each iteration.

Table 4.1: Sensitivity study of the effect of $\gamma_g = \eta\gamma_m$ on the overall convergence rate in P6 (shielding enforced).

Feedback	Power Level	η							
		0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
TH	140%	16	14	14	14	**12**	**12**	14	14
	100%	**12**	**12**	**12**	**12**	**12**	13	14	14
TC	140%	18	15	**13**	**13**	14	15	16	16
	100%	13	**12**	**12**	13	13	13	14	16
TCE	140%	16	15	14	**13**	**13**	**13**	**13**	14
	100%	14	13	**12**	**12**	**12**	**12**	**12**	**12**

In the second study, the shielding calculation is triggered by the change in fuel temperature between iterations. γ_m is again used to estimate the NOPC parameters at the same outer iteration. For the outer iteration where the shielding calculation is not performed, the γ_g used to estimate the partial convergence is varied in a range incorporating $\bar{\gamma}$. Table 4.2 summarizes the total outer iteration number to converge the problem. It can be seen that using $\bar{\gamma}$ to estimate the partial convergence results in an optimal value for the total number of iterations so long as it is used when no self-shielding calculation is performed–or equivalently, the multiphysics solution is close to the converged.

Table 4.2: Sensitivity study of the effect of $\gamma_g = \eta\bar{\gamma}$ on the overall convergence rate in P6.

Feedback	Power Level	η							
		0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
TH	140%	**12**	**12**	**12**	**12**	**12**	**12**	**12**	**12**
	100%	**13**	**13**	**13**	**13**	**13**	**13**	**13**	**13**
TC	140%	**13**	**13**	**13**	**13**	14	14	14	14
	100%	**12**	**12**	**12**	**12**	**12**	13	13	13
TCE	140%	**12**	**12**	**12**	**12**	**12**	**12**	**12**	**12**
	100%	**12**	**12**	**12**	**12**	**12**	**12**	13	13

4.3.6 Summary

This section reviewed the effects of partial convergence and relaxation on a coupled Picard iteration. The case was made that it is more efficient and robust to rely on partial convergence to stabilize the Picard iteration, rather than relaxation. For simple problems, it was illustrated that (1) an optimal partial convergence (or relaxation) exists, and (2) that this optimum is problem-dependent and can be quite sensitive. From work in Chapter 3 we propose to compute an optimal WS ratio to achieve nearly-optimal partial convergence in a problem-dependent manner. The calculation of this NOPC parameters hinges on the ability to calculate the localized feedback intensity γ, and the global feedback intensity γ_g. Calculating γ and γ_g is non-trivial, and if it is not done sufficiently well, then the NOPC of the CMFD solution and optimal global convergence of the iteration scheme will not be achieved. We developed a few relatively simple procedures to estimate γ. Then we presented numerical calculations to assess sensitivities and provide overall recommendations as to how to calculate γ and γ_g so that the multiphysics iteration scheme is stable and converges nearly optimally (i.e., ensures that it is robust). We demonstrated that for canonical PWR problems, the convergence of the coupled problem can be achieved in as many iterations as the uncoupled problem for a wide range of operating conditions. In the next section, we apply the procedure developed here to several PWR applications from pin cell to full-core configuration to demonstrate its efficacy for realistic problems.

4.4 Numerical Results for Realistic Problems

Now the numerical results of the new robust-relaxation free scheme are presented. The performance of the two NOPC-CMFD methods is compared to the CMFD methods currently used in MPACT: MGPI, MSED, and MSEDL. The MSEDL is a variant of MSED that simply limits the amount of convergence of the diffusion solver in a fixed way [32] by loosening the convergence criteria and reducing the aggressiveness of λ_s. This variant was developed because MSED was too efficient at converging the CMFD problem, resulting in less stable and even unstable multiphysics iterations. No relaxation is used for NOPC-CMFD methods. For MGPI, MSED, and MSEDL, the default relaxation factor in MPACT, $\beta = 0.5$, is applied when the pin-power is transferred to the TH solver.

The convergence criteria, the maximum number of power iterations, and the relaxation factor used for the iteration scheme with these solvers are summarized in Table 4.3. As introduced in Algorithm 5, L_{MG} is the number of iterations performed on multi-group system for all the solvers. L_{GR} is the number of power iterations performed on the 1GCMFD system per MGCMFD per fixed source iteration. For MGPC and MSPC, two values are specified L_{MG}, respectively. The first is the number of power iterations performed before estimating the feedback intensity and partial convergence. The second is the number of power iterations for estimating the partial convergence. More power iterations are performed before estimating the partial convergence to let the intermediate flux get closer to the eigenvector of the dominant k-eigenvalue.

Table 4.3: Convergence specification and relaxation for cases of the CMFD solvers.

CMFD Solver	L	L_{MG}	L_{GR}	k_{tol}	r_{tol}	λ_s	Relaxation
MGPI	L_{MG}	20	-	10^{-6}	10^{-2}	$2/3$	0.5
MSED	$L_{MG}L_{GR}$	20	5	10^{-6}	10^{-2}	adaptive	0.5
MSEDL	$L_{MG}L_{GR}$	20	5	10^{-4}	5×10^{-2}	$2/3$	0.5
MGPC	L_{MG}	8/5	-	10^{-6}	10^{-3}	nearly-optimal	-
MSPC	$L_{MG}L_{GR}$	5/4	5	10^{-6}	10^{-3}	nearly-optimal	-

Since our goal is to have the convergence of the multiphysics problems behave in the same way as the pure transport problem, the number of outer iterations obtained from the no feedback case is considered as the practical lower bound for the convergence of the multiphysics problem. Therefore, we use the number of outer iterations to converge the problem, when no feedback is present and the MSED solver is used with the tight convergence specified for the CMFD solution, as the reference result in most problems.

4.4.1 3D Pin Problem

We first assess the performance of the NOPC-CMFD method, MGPC, against the current CMFD method in MPACT for a simple problem: 3D P1. The variants of the MSED solver are not yet investigated at this point.

For the simple single rod problem, the current iteration scheme is degraded and will fail to converge when feedback is present and becomes stronger. As shown in Figure 4.9, when no feedback is present, and the CMFD solution is fully converged each outer iteration, it requires **16** outer iterations to reach the convergence criteria for the fission source residual, $\epsilon_f = 10^{-6}$. This result is considered as a reference for the number of outer iterations of the classical CMFD method without feedback. The fission source residual decays quickly before the 4^{th} outer iteration after which the asymptotic convergence rate (i.e., the spectral radius) is observable. At the beginning of the simulation, the reduction of the residual is determined by the fast-decaying components.

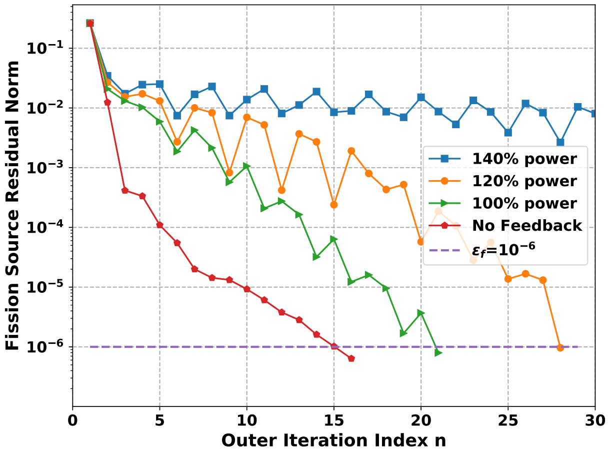


Figure 4.9: Evolution of the fission source residual with the outer iteration for 3D P1 cases. The CMFD solutions are fully converged each outer iteration. The feedback is from TH and the relaxation factor is 0.5. The oscillation of the residual norm is caused by, and addressed with, relaxation.

When feedback is present, and the CMFD solutions are fully converged each outer iteration, the fission source residual converges much slower than that in the case without any feedback. The fission source residual in the cases with feedback oscillates with the outer iteration index increasing. This indicates that the spectral radius of the iteration scheme in these problems is complex. The result supports the theoretical prediction in [39] that when relaxation is applied for the pin powers, the spectral radius of the iteration scheme could be complex. The results also illustrate that even for the simple pin problem, the CMFD solutions should never be fully converged. This same convergence behavior can also be observed with Monte Carlo transport [26]; that suggests the fundamental issue is due to over-converging the neutronics between updates from the thermal hydraulics–rather than an issue with the discretization of the transport or acceleration operators.

In practical simulations, we do not always try to converge the CMFD solutions fully. Therefore, the iteration scheme with the default MGPI method is observed to be stable in most cases, as shown in Table 4.4. However, the partial convergence of the MGPI solution is determined by a fixed specification, as shown in Table 4.3. Therefore, in some cases, the partial convergence of the MGPI is not optimal. This is observed in the MGPI result for the 3D P1 case with TCE feedback at 140% rated power, or the case with TC feedback at 20% power that has 19 or 18 iterations, respectively. The results indicate that the partial convergence for the MGPI may be too loose for the low-power 3D P1 cases and too tight for the high-power 3D P1 cases, but otherwise well-tuned to the 100% power condition.

In contrast, the MGPC method adjusts the partial convergence of the CMFD solution according to the feedback intensity of the problem. The method, therefore, allows the CMFD solutions to be partially converged nearly-optimally. Moreover, the range of the number of total outer iterations is [15,17]. Compared to the total outer iterations of the classical CMFD, the potential overhead from being *nearly* optimal is at most 1 iteration.

Table 4.4: Convergence behavior of different iteration schemes in 3D P1 with different total powers.

FDB	CMFD Solver	Power Level (%)													
		20	40	60	80	100	120	140	20	40	60	80	100	120	140
		Outer Iterations							**Shielding Calculations**						
TH	MGPI	16	16	16	16	16	16	17	3	3	5	5	6	7	9
	MGPC	16	16	16	16	16	16	17	3	4	4	4	4	4	5
TC	MGPI	18	15	15	16	16	16	17	3	5	5	5	7	8	10
	MGPC	15	15	15	16	16	16	16	3	5	5	5	6	6	6
TCE	MGPI	15	16	16	16	16	17	19	3	4	5	6	7	8	11
	MGPC	15	16	16	16	16	17	17	2	4	5	5	5	5	5

The number of shielding calculations performed in each case is also presented in Table 4.4. The shielding calculation is triggered if the maximum of the pin-averaged temperature change between two successive iterations is larger than 5 kelvin. Therefore, the number of shielding calculations is a metric to indicate how fast the state variables (e.g., temperature) converge. It can be seen that the number of shielding calculations performed in the MGPC cases are the same as or smaller than the number of shielding calculations in the MGPI cases.

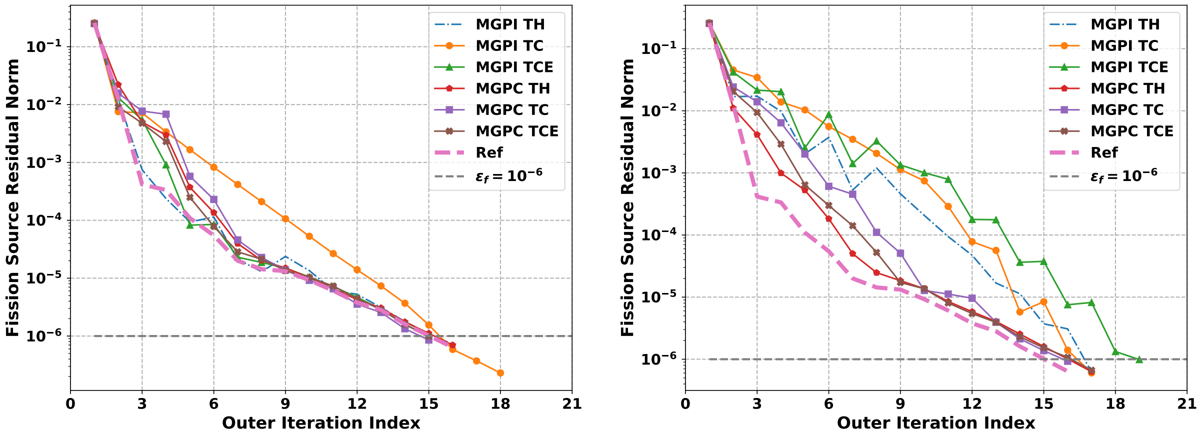
Table 4.5: Run time of MPACT with different iteration schemes in 3D P1 with different total powers.

FDB	CMFD Solver	Power Level (%)						
		20	40	60	80	100	120	140
TH	MGPI t_1 (min)	4.4	4.4	4.7	4.8	4.8	5.1	5.5
	MGPC t_2 (min)	4.5	4.5	4.5	4.5	4.4	4.3	4.9
	$(t_1 - t_2)/t_1$	2%	2%	-4%	-6%	-8%	-16%	-11%
TC	MGPI t_1 (min)	4.8	4.5	4.5	4.9	5.2	5.4	6.0
	MGPC t_2 (min)	4.2	4.6	4.6	4.9	5.1	5.1	5.0
	$(t_1 - t_2)/t_1$	-13%	2%	2%	0.0%	-2%	-6%	-16.7%
TCE	MGPI t_1 (min)	4.1	4.4	4.6	4.8	5.1	5.7	6.7
	MGPC t_2 (min)	4.2	4.7	4.9	4.9	5.0	5.2	5.2
	$(t_1 - t_2)/t_1$	2%	7%	7%	2%	-2%	-9%	-22%

In terms of the run time, for the cases with power between 20% and 120% rated power, the MGPI and MGPC have comparable run time. The magnitudes of the differences in the run time are less than 10% of the run time of the MGPC cases. When power is larger than 100% rated power, the MGPC cases run faster than the MGPI cases. This happens for 3 reasons: compared to the MGPI cases, (i) less work is being done in the CMFD solve, (ii) there are fewer outer iterations, and (3) there are fewer shielding calculations. This allows the run time to be reduced by as much as 22%. It can also be observed that with increasing power, the total run time of both methods is also increasing–this is because more shielding calculations are being performed. However, the time increase of the cases with power from 20% to 140% rated power is less for the MGPC method than the MGPI method. It can also be observed that for the MGPC cases, the run time of the case with 120% rated power is shorter compared to that with 20% rated power, though more shielding calculations are performed in the 120% rated power case when the feedback is just TH. The reason is again that less work is done to solve the CMFD equations for the 120% rated power case due to a more loose partial convergence determined by Eq. (3.40a).

Figures 4.10 illustrates the convergence behavior of the fission source residual in the cases with 20% rated power and 140% rated power. Here it is observed that the convergence behavior of MGPC is relatively insensitive to the form of feedback, and is as good as or better than MGPI for various power levels.

The accuracy of the simulations is compared using the data in Table 4.6 and Table 4.7. These tables show the results of the eigenvalue, k, or critical boron concentration, C_B, the maximum pin-averaged temperature, ΔT, and the position, P_T that indexes the pin-cell location at which the maximum temperature occurs for the MGPI and MGPC cases with different power levels. For a



(a)

(b)

Figure 4.10: Comparison of NOPC-CMFD solvers to other CMFD methods for the convergence in the 3D P1 cases with 20% power and 140% power. The convergence behaviors of the fission source residual for cases with different solvers and feedback from different physics are compared. (a) shows that the MGPC method has comparable performance to the MGPI method in low-power cases, and (b) shows that the MGPC method is more stable than MGPI in the cases with high power.

good acceleration method we desire that these quantities do not vary outside the convergence criteria, and that is what is observed here. There are slight discrepancies that arise from the resonance self-shielding calculation not being performed at each iteration. If the resonance self-shielding calculation is performed for every outer iteration, then the comparison is consistent, and the differences would be within convergence.

Table 4.6: Results of 3D P1 cases with different iteration schemes with feedback from TH.

	CMFD Solver								
Power	MGPI			MGPC			Difference		
Level (%)	k (pcm)	T_{max} (K)	P_T	k (pcm)	T_{max} (K)	P_T	$\lvert\Delta k\rvert$ (pcm)	$\lvert\Delta T\rvert$ (K)	$\lvert\lvert\Delta P_T\rvert\rvert_1$
100	1.165904	1077.54	(33)	1.165903	1078.10	(32)	0.1	0.57	1
120	1.162287	1269.37	(32)	1.162279	1269.35	(31)	0.8	0.02	1
140	1.158604	1472.31	(31)	1.158604	1473.01	(33)	0.1	0.70	2

Since the emphasis of the work is to show the efficiency and robustness of the relaxation-free method, in the following sections, the simulated results C_B, k, T_{max} and P_T are not presented for brevity. The observations are similar to what has been observed on this single rod problem. Namely that the method does not lead to significant differences in the solution, although slight, negligible

Table 4.7: Results of 3D P1 cases with different iteration schemes with feedback from TC and TCE.

		CMFD Solver								
FDB	Power Level (%)	MGPI			MGPC			Difference		
		C_B (ppm)	T_{max} (K)	P_T	C_B (ppm)	T_{max} (K)	P_T	$\lvert\Delta C\rvert$ (ppm)	$\lvert\Delta T\rvert$ (K)	$\lvert\lvert\Delta P_T\rvert\rvert_1$
TC	100	3390.97	1102.57	(44)	3390.99	1102.48	(44)	0.02	0.09	0
	120	3363.50	1301.49	(45)	3363.51	1301.96	(45)	0.01	0.47	0
	140	3335.47	1510.94	(45)	3335.51	1511.74	(45)	0.04	0.79	0
TCE	100	2993.43	1050.94	(42)	2993.44	1051.21	(43)	0.01	0.27	1
	120	2946.55	1239.75	(43)	2946.59	1240.75	(43)	0.04	1.00	0
	140	2902.40	1438.92	(43)	2902.45	1439.88	(43)	0.05	0.97	0

differences may result from not performing the subgroup self-shielding calculation every outer iteration.

4.4.2 Single Assembly Problem

In this section, P6 is used to investigate and compare the performance of different CMFD methods in the presence of feedback. As mentioned previously, the total number of outer iterations to converge cases without feedback using the classical CMFD solver is **12**. This is taken to be the reference lower bound.

Table 4.8 summarizes the performance of the different methods for the P6 cases with various combinations of power and feedback. It is observed that most MGPI cases are stable, and the number of outer iterations to converge is similar to that of the classical CMFD. However, the performance of MGPI is significantly degraded, and the number of outer iterations to converge is much larger than 12, when the power is 140% rated power. The MGPI performance in the P6 case is also degraded if the feedback is from TC with 20% rated power.

The variants of the MSED solver are much more efficient than the MGPI and MGPC solver in the steady-state calculations without feedback. For the 20% power case, where the feedback is not strong, the MSED method converges in the same number of outer iterations as the reference case. However, the efficient MSED solver makes the iteration scheme less stable as the feedback intensity increases. For the P6 cases with 140% rated power, the MSED method requires more than 25 outer iterations to converge. The MSEDL method is observed to be more robust than the MSED method in cases that have greater than 100% rated power–which is by design. Nevertheless, this method is still not *always* robust, nor optimal, in multiphysics simulations. This is due to the fact that the imposed convergence penalty on MSEDL compared to MSED was obtained empirically

Table 4.8: Convergence behavior of different iteration schemes in P6 with different total powers.

FDB	CMFD Solver	Power Level (%)									
		20	60	100	120	140	20	60	100	120	140
		Outer Iterations					**Shielding Calculations**				
TH	MGPI	13	13	13	13	15	2	5	6	6	7
	MGPC	13	13	13	13	12	3	5	5	4	4
	MSED	12	13	15	21	27	3	5	7	10	16
	MSEDL	16	15	14	13	14	4	4	5	5	7
	MSPC	12	12	13	13	12	2	3	4	5	6
TC	MGPI	16	13	12	14	15	2	6	7	8	9
	MGPC	12	12	12	13	13	4	5	6	7	7
	MSED	12	13	17	25	37	3	6	8	13	22
	MSEDL	17	16	16	19	22	4	5	8	11	14
	MSPC	12	12	12	12	12	3	5	5	6	7
TCE	MGPI	12	12	13	15	17	3	5	6	7	10
	MGPC	13	13	13	13	12	3	5	5	4	4
	MSED	12	16	22	32	49	4	6	11	17	29
	MSEDL	12	12	14	17	21	4	5	7	9	11
	MSPC	12	12	12	12	12	3	5	5	6	7

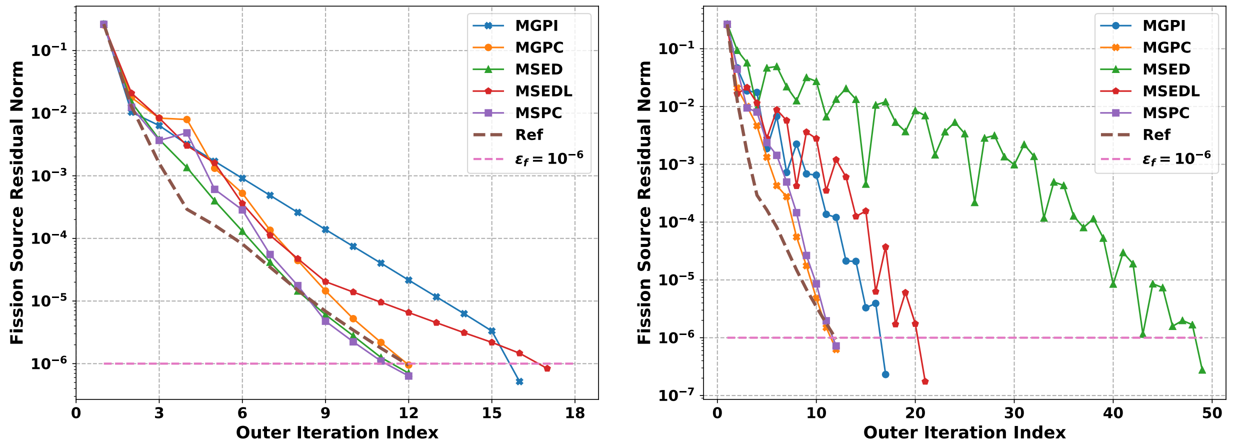
by numerical experimentation on a few problems. For the cases of 20% rated power with feedback from TH and TC, the MSEDL method may take more than 16 outer iterations to converge. When the power is larger than 100% rated power and the feedback is from TC or TCE, the iteration scheme converges the P6 cases in more than 17 outer iterations. The performance of the MSEDL method is only observed to be nearly-optimal if the power of the case is 120% rated power.

Contrasting these observations with the results from the NOPC-CMFD solvers in MPACT, MGPC or MSPC, it is observed that convergence is nearly optimal in all cases. The MGPC or MSPC methods converge all P6 cases in at most 13 outer iterations, and in most cases it requires only 12. Therefore, both MGPC and MSPC have comparable performance to the classical CMFD for the P6 cases. This demonstrates that the NOPC-CMFD can efficiently converge the CMFD solution in a problem-dependent way, and is robust in the sense that one can expect the same number of iterations to converge irrespective of the feedback mechanisms or intensity.

In terms of the number of shielding calculations, it can be observed that for the MGPC and MSPC method, the number of shielding calculations performed in most cases is the same. When the number of shielding calculations is not the same, the difference in the number of shielding calculations is no larger than 2. Among all the methods investigated, the number of shielding calculations taken for the two NOPC-CMFD methods is close to the smallest value. We wish to

Table 4.9: Run time comparison of different CMFD methods in P6 with different total powers.

FDB	CMFD Solver	Power Level (%)									
		20	60	100	120	140	20	60	100	120	140
		Run Time (min)					**Time Difference**				
TH	MGPI	4.4	5.1	5.8	6.0	6.9	–	–	–	–	–
	MGPC	4.8	5.0	5.1	4.9	4.6	9%	-2%	-12%	-18%	-33%
	MSED	3.5	4.1	4.9	6.8	9.2	-20%	-20%	-16%	13%	33%
	MSEDL	4.5	4.3	4.3	4.0	4.7	2%	-16%	-26%	-33%	-32%
	MSPC	3.3	3.5	3.9	4.1	4.1	-25%	-31%	-33%	-32%	-41%
TC	MGPI	5.8	5.4	5.5	6.5	7.2	–	–	–	–	–
	MGPC	4.8	4.8	5.0	5.5	5.5	-17%	-11%	-9%	-15%	-24%
	MSED	3.5	4.2	5.5	8.2	12.5	-40%	-22%	0%	26%	74%
	MSEDL	4.8	4.7	5.3	6.6	7.8	-17%	-13%	-4%	2%	8%
	MSPC	3.5	3.9	3.9	4.1	4.3	-40%	-28%	-29%	-37%	-40%
TCE	MGPI	4.7	5.1	5.9	7.0	8.3	–	–	–	–	–
	MGPC	4.4	4.8	4.9	5.0	5.1	-6 %	-6%	-17%	-29%	-39%
	MSED	3.8	5.1	7.4	10.8	17	-19%	0%	25%	54%	105%
	MSEDL	3.8	3.9	4.8	5.9	7.3	-19%	-24%	-19%	-16%	-12%
	MSPC	3.6	4.0	4.0	4.2	4.4	-23%	-22%	-32%	-40%	-47%



(a) **(b)**

Figure 4.11: Variation of the fission source residual with the outer iteration for P6 cases. The convergence behavior of the fission source residual for cases with different solvers and feedback from different physics are compared. (a) shows the results of cases with 20% rated power with feedback from TC, while (b) shows the results of cases with 140% rated power with feedback from TCE.

note this since the computational cost of the shielding calculation is similar to that of performing a CMFD accelerated transport iteration, and the number of times it is executed can strongly influence the total run time.

Table 4.9 shows the run time of each case, and compares the run time of the CMFD methods with the run time of the current MGPI method. The run time of the MGPC method is a bit longer than that of the MGPI method in the case with 20% rated power and the feedback from TH. However, the overhead of the run time of the MGPC method is relatively small. For all the other cases, the MGPC method takes less time than the MGPI method. At most, the MGPC method was able to reduce the run time by 33% for the 140% power case.

The MSED and MSEDL cases illustrate the deficiencies demonstrated earlier. MSED generally more tightly converges the CMFD solutions than MGPI and this results in degraded performance at high powers. The MSEDL variant does better at high powers, but low power cases converge more slowly, exposing the fact that the amount of partial convergence needed is problem dependent and this is not handled by MSEDL. However, incorporating the NOPC with MSED is clearly demonstrated to be a good strategy as the MSPC method has the fewest number of iterations and run times in every case, with run time reductions ranging from about 20% to nearly 50%.

Figures 4.11 show the convergence behavior of the fission source residual in the P6 case with 20% rated power and feedback from TC, and the P6 case with 140% rated power with feedback from TCE. Compared to the fission source residual in the reference (no feedback) case, it can be seen that the fission source residual of the MGPC cases and MSPC cases are larger at the first several outer iterations. The reason is that the state variables (and cross sections) are not well converged and thus vary considerably for the first several iterations. However, after the state variables and cross sections become close to their converged values, it can be seen that the fission source residual converges much faster and even faster than that in the reference case, indicating that the fission source is converged nearly-optimally. The slow convergence of MSEDL and MGPI in the problem with 20% rated power is caused by the loose partial convergence of CMFD solutions. As observed in Figure 4.11a, the MGPI and MSEDL converge the fission source asymptotically at a rate much slower than that of MGPC, MSPC, and the reference cases. The slow convergence of MSEDL, MSED, and MGPI in the problem with 140% rated power, is caused by converging the CMFD solutions too tightly for the feedback. It can also be seen that the fission source residual oscillates in many of the conventional schemes. This observation is similar to what was observed for the fully converged CMFD solver in Figure 4.9.

4.4.3 Effect of Iterative T/H Solutions

So far, all the numerical results presented use the simplified TH model for the TH solver, where the TH solutions can be understood to be fully converged and contain no iterative errors. This is an assumption that has thus far been applied to the analytic work in Chapter 3 and the numerical results for more practical problems where we demonstrated that the theory translates well to practical benefits for problems of interest.

To investigate the performance of the NOPC-CMFD methods in simulations with more complex TH models that rely on iterative solutions as well, the CTF [90] is coupled with MPACT. The CTF solutions are obtained through iteration and are not fully converged. Since the tight convergence criteria for the fission source residual is used, the tolerance for converging the CTF solution is determined by

$$\epsilon_i = \max(1e-6, \epsilon_0 res_f)\,, \tag{4.9}$$

where res_f is the residual of the fission source, and ϵ_0 is the input tolerance for converging the CTF solutions.

The results are shown in Table 4.10. It can be seen that the cases with the NOPC-CMFD method, MGPC or MSPC, converge faster than the MGPI cases or MSEDL cases–a result consistent with the simplified T/H solver. The number of outer iterations in the NOPC-CMFD cases is around 12 and no more than 13, while the outer iterations for the MGPI cases or MSEDL cases are close to the reference value 12 when the power is neither too low nor too high. Fewer shielding calculations are performed for the MGPC or MSPC cases than for the MGPI cases or MSEDL cases. The result indicates that the NOPC-CMFD solvers can be used in simulations with more accurate and complex TH solvers, *provided there is sufficient convergence of the T/H solution before updating cross sections*; where "sufficient" is defined by Eq. (4.9).

4.4.4 3×3 Core Problem

The VERA core physics benchmark problem 4 (P4) is now used to investigate how MPACT with the NOPC-CMFD solvers performs in problems with control rods and Pyrex (PY) burnable neutron poison rods inserted [98]. The layout of the P4 assemblies is shown in Figure 4.12. The control rod bank, i.e., the hybrid silver-indium-cadmium (AIC)/B_4C rod cluster control assembly (RCCA), is located in the central assembly, and the Pyrex rods are in the 2.6% enriched region.

In this section, the position of the RCCA tip is provided in the number of steps withdrawn for the bank, with the step size being 1.5875 cm. When the RCCA is fully inserted, i.e., the RCCA movement is at 0 steps withdrawn (SWD), the distance from the tip to the top of the bottom core plate is 16.285 cm.

Table 4.10: Convergence behavior of different iteration schemes in P6 with different total powers with CTF as TH Solver.

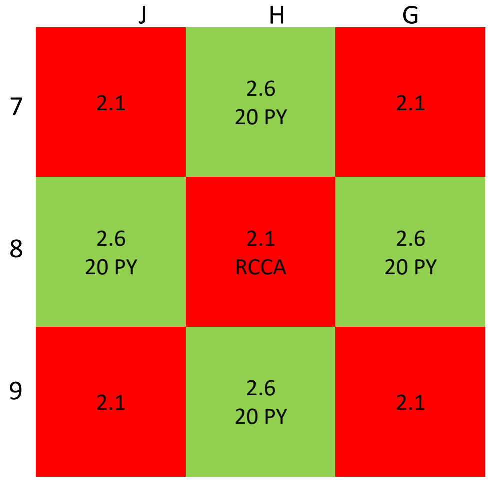
FDB	CMFD Solver	Power Level (%)									
		20	60	100	120	140	20	60	100	120	140
		Outer Iterations					**Shielding Calculations**				
TH	MGPI	13	13	14	12	13	3	4	5	6	6
	MGPC	13	13	13	12	12	3	4	4	4	4
	MSEDL	17	16	15	16	19	4	5	7	8	9
	MSPC	12	12	13	12	12	2	3	4	5	5
TC	MGPI	16	12	12	14	15	3	5	6	7	8
	MGPC	12	12	12	13	13	3	5	6	6	6
	MSEDL	17	16	15	16	19	4	5	7	8	9
	MSPC	12	12	12	12	13	2	5	5	5	6
TCE	MGPI	12	13	12	14	16	3	5	6	7	8
	MGPC	12	12	12	12	12	2	5	5	5	6
	MSEDL	12	13	14	15	18	3	5	6	7	9
	MSPC	12	12	12	12	12	2	4	5	5	5

The power, feedback, and the rod position of the cases studies for this problem are summarized in Table 4.11. The cases of 20% and 100% rated power are used to show the performance of the methods in the problems of both low and normal power. The rod positions analyzed are from 23 SWD to 92 SWD at an increment of 23 SWD. The problem is simulated in quarter symmetry. While the authors acknowledge that deeply inserted RCCA's at the nominal power is never the case for operation in a PWR due to the rod insertion limits, the simulation is valuable for assessing the robustness of NOPC-CMFD across broader and more challenging conditions. An example of the axial power profile is shown in Figure 4.13. It can be seen that the P4 cases are highly heterogeneous along the axial direction.

Table 4.11: Specification of the P4 cases.

Power Level (%)	Feedback Form	Control Rod Position (step)
20,100	TH, TC, TCE	23,46,69,92

The MSED solver is not investigated in this section due to its slow convergence in problems with feedback. The number of outer iterations that MPACT requires with the classical CMFD method to converge P4 with different RCCA positions are **13**, **14**, **13**, and **13**, respectively. These are taken as the reference results in Table 4.12 and Table 4.14.



	J	H	G
7	2.1	2.6 20 PY	2.1
8	2.6 20 PY	2.1 RCCA	2.6 20 PY
9	2.1	2.6 20 PY	2.1

Figure 4.12: Problem 4 assembly, poison and control layout (Figure P4-1 in Ref. [98]).

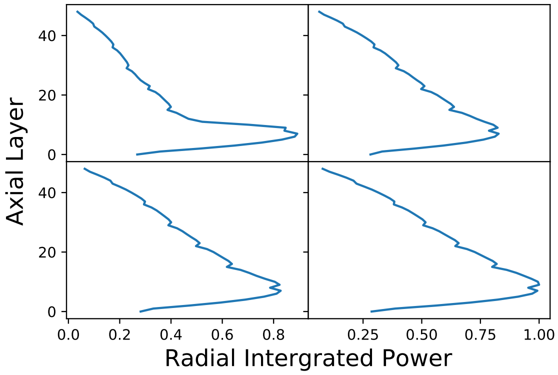


Figure 4.13: An example of the axial power profile of P4. The top left is inserted with RCCA. The RCCA is withdrawn 46 steps from the core. The power is 100% rated power, and the feedback is from TCE.

Table 4.12 and Table 4.14 also present the number of outer iterations to converge these cases and the number of shielding calculations performed.

For the cases of 20% rated power, it can be seen that the NOPC-CMFD solvers make the iteration scheme converge the problem with the same number of, or fewer outer iterations, than the reference cases. The current CMFD method MGPI, requires more outer iterations to converge compared to the reference cases–when feedback is from TH or TC. The MSEDL method takes 19 outer iterations to converge some cases indicating, again, the shortcoming of using a predetermined limited convergence factor. In terms of the number of shielding calculations, the different methods require a different number of shielding calculations, but the differences are small.

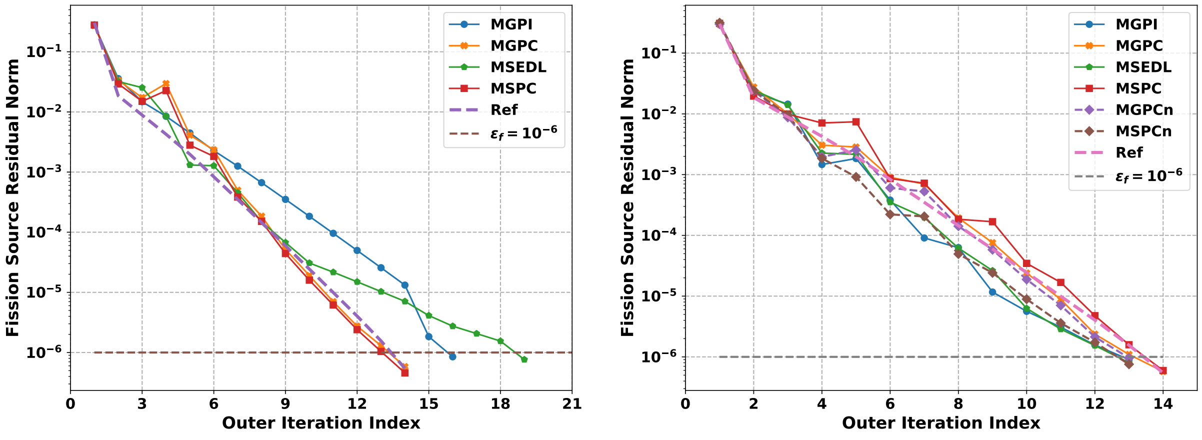
Figure 4.14a shows the convergence behavior of the fission source of the P4 problem with the RCCA withdrawn to 46 steps and with only TH feedback. It can be seen that the MGPI and

Table 4.12: Convergence behavior of iteration schemes in P4 cases with different control rod positions (20% rated power).

FDB	CMFD Solver	Rod Position							
		23	46	69	92	23	46	69	92
		Outer Iterations				Shielding Calculations			
–	**Ref**	13	14	13	13	–	–	–	–
TH	MGPI	16	16	15	14	4	6	5	5
	MGPC	13	14	13	13	4	5	5	5
	MSEDL	19	19	17	15	4	5	5	5
	MSPC	13	14	13	13	4	5	5	5
TC	MGPI	16	16	15	14	4	5	5	5
	MGPC	13	14	13	13	5	5	5	5
	MSEDL	19	19	17	14	4	6	5	4
	MSPC	13	14	13	13	5	5	5	5
TCE	MGPI	13	14	14	13	4	4	4	4
	MGPC	13	13	13	13	3	5	5	4
	MSEDL	14	13	14	13	4	4	4	4
	MSPC	13	13	13	13	3	3	4	3

Table 4.13: Run time comparison of iteration schemes in P4 cases with different control rod positions (20% rated power).

FDB	CMFD Solver	Rod Position							
		23	46	69	92	23	46	69	92
		Run Time (min)				**Time Difference**			
TH	MGPI	17.5	19.0	16.6	15.3	–	–	–	–
	MGPC	13.8	15.1	13.8	13.5	-21%	-21%	-17%	-12%
	MSEDL	13.5	14.0	12.8	11.6	-23%	-26%	-23%	-24%
	MSPC	10.2	11.2	10.6	10.6	-42%	-41%	-36%	-31%
TC	MGPI	17.6	18.4	16.6	15.3	–	–	–	–
	MGPC	14.4	15.1	13.8	13.6	-18%	-18%	-17%	-11%
	MSEDL	13.5	14.5	12.8	10.6	-23%	-21%	-23%	-31%
	MSPC	10.7	11.2	10.7	10.7	-39%	-39%	-36%	-30%
TCE	MGPI	13.6	14.9	14.5	13.7	–	–	–	–
	MGPC	12.8	13.9	13.7	13.2	-6 %	-7%	-6 %	-4%
	MSEDL	11.0	10.3	10.9	10.3	-19%	-31%	-25%	-25%
	MSPC	10.0	10.0	10.5	10.0	-26%	-33%	-28%	-27%



(a) **(b)**

Figure 4.14: Comparison of NOPC-CMFD solvers to other CMFD methods for P4 cases. The convergence behavior of the fission source residual for cases with different solvers and feedback from different physics are compared. (a) shows that the results of cases with 20% rated power with feedback from TH. The RCCA is withdrawn 46 steps. (b) shows that the results of cases with 100% rated power with feedback from TC. The RCCA is withdrawn 92 steps.

MSEDL converge the fission source asymptotically more slowly than the classical CMFD, MGPC, and MSPC. These results indicate that the partial convergence of the MGPI solutions and the MSEDL solutions are not tight enough for the iteration scheme. The MGPC and MSPC methods have more tightly converged the CMFD solutions, therefore they have similar performance to the classical CMFD. The results agree with what has been observed in the low-power case of the 3D P1 and P6 problems, and the predictions of the analysis in Chapter 3.

The run time of the cases with 20% rated power for each CMFD method and run time difference are presented in Table 4.13. The run time difference of the CMFD methods and the MGPI method is shown in the percentage of the run time of the MGPI method. It can be seen that the MGPC cases run faster than the MGPI cases. For low-power cases, MSEDL runs faster than the MGPI method because of its lower CMFD computational cost. However, it converges slowly due to the loose partial convergence of the CMFD solutions. Consequently, the run time reduction is not as large as that of the MSPC method. Finally, it can be observed that the run time of MSPC is at least 26% less and at most 42% less than the run time of MGPI. This result is consistent with the P6 results indicating that multiple assemblies and partially inserted control rods do not have a substantial direct effect on the convergence behavior of NOPC.

For the cases of 100% rated power, as shown in Table 4.14, it can be found that the MGPI converges much faster, and requires a similar number of outer iterations as the reference case. The MSEDL method in general, does not perform as well as the MGPI method. When the RCCA is withdrawn 23 steps, and the feedback is from TH only, the MSEDL method requires 17 iterations

Table 4.14: Convergence behavior of iteration schemes in P4 cases with different control rod positions (100% rated power).

FDB	CMFD Solver	Rod Position							
		23	46	69	92	23	46	69	92
		Outer Iterations				Shielding Calculations			
–	**Ref**	13	14	13	13	–	–	–	–
TH	MGPI	13	13	13	13	5	7	7	7
	MGPC	13	13	14	13	5	7	7	6
	MSEDL	17	13	15	13	6	8	8	7
	MSPC	13	14	14	13	6	6	6	6
TC	MGPI	13	14	13	13	6	7	7	6
	MGPC	13	14	14	14	6	7	7	6
	MSEDL	13	16	14	13	5	7	7	7
	MSPC	14	14	14	14	6	7	7	8
TCE	MGPI	13	14	13	13	7	6	8	6
	MGPC	13	13	13	13	6	6	6	6
	MSEDL	14	15	13	13	6	8	7	6
	MSPC	14	13	13	13	6	6	6	6

to converge. The MGPC and MSPC have comparable performance to the MGPI in terms of the total outer iterations. Though the MGPI method performs slightly better, it is not guaranteed to do so for every condition. Compared to the reference cases, the MGPC method takes no more than 1 outer iteration to converge the P4 cases and again remains insensitive to the type of or magnitude of the feedback.

Figure 4.14b shows the convergence behavior of the fission source of the P4 problem with RCCA withdrawn 92 steps, and feedback is TC. Here it is observed that the MGPI and MSEDL converge the fission source asymptotically at a similar rate to the classical CMFD. These results indicate that the partial convergence of the MGPI solutions and the MSEDL solutions are near-optimal. However, that happens only for this case, and is not generally true. The MGPC and MSPC methods converge asymptotically faster than the classical CMFD, and the MGPC and MSEDL. They take one more outer iteration to converge because of the partial convergence of MGPC and MSPC for the initial iterations when γ for the problem is not well determined, and the WS factor is not yet near its optimum.

To verify this claim, $\gamma_e = 1.2 \times 10^{-3}$ rather than $\gamma_e = 1.5 \times 10^{-3}$ is used as the initial estimate for γ before the 4^{th} outer iteration. The convergence behavior of the fission source residual for these two methods are labeled as MGPCn and MSPCn. It can be seen that the two methods take

13 outer iterations to converge. Therefore, the MSPCn can converge the fission source similarly to the classical CMFD. This result also illustrates a drawback of this method: γ_e is still a user-specified value. In this thesis, it is recommended that γ_e be set to 1.5×10^{-3} for PWRs. In some cases, the partial convergence estimated by this value may not be near-optimal, but it will degrade the performance of the method only slightly. Ultimately, though we suggest that the overall performance and robustness of the NOPC-CMFD scheme is far less sensitive to γ_e than it is to the relaxation parameter of the traditional Picard iteration scheme. Therefore, the NOPC-CMFD method is still advantageous despite this drawback.

In terms of run time, as shown in Table 4.15, when the feedback is from TH and TC, the MGPC has a comparable run time to MGPI because the MGPI cases converge nearly-optimally in these cases. When the feedback is from TCE, i.e., the feedback becomes stronger, the MGPC method performs much better. The run time is at least 6% less than the run time of MGPI and can be as much as 17% less. The run time of MSEDL and MSPC is comparable in most cases, differences in run time are again mainly caused by the differences in the outer iterations and shielding calculations.

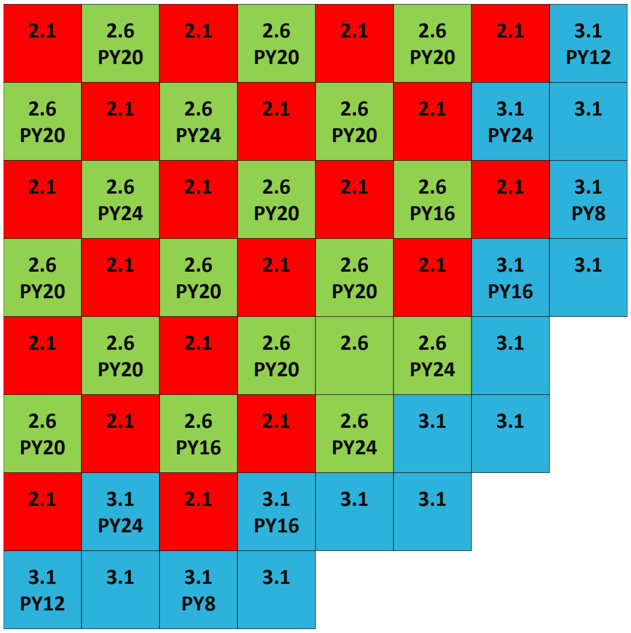
Table 4.15: Run time comparison of iteration schemes in P4 cases with different control rod positions (100% rated power).

FDB	CMFD Solver	Rod Position							
		23	46	69	92	23	46	69	92
		Run Time (min)				**Time Difference**			
TH	MGPI	13.9	15.1	15.8	14.5	–	–	–	–
	MGPC	14.1	15.7	15.6	15.1	1%	4%	-1%	4%
	MSEDL	13.3	12	13.1	11.4	-4%	-21%	-17%	-21%
	MSPC	11.2	11.8	11.8	11.2	-19%	-22%	-25%	-23%
TC	MGPI	15.4	16.6	15.9	14.7	–	–	–	–
	MGPC	14.5	15.9	15.8	15.1	-6 %	-4%	-1%	3%
	MSEDL	10.5	13.2	12	11.5	-32%	-20%	-25%	-22%
	MSPC	11.7	12.3	12.3	12.7	-24%	-26%	-23%	-14%
TCE	MGPI	17.1	17.2	17.5	15.7	–	–	–	–
	MGPC	14.4	14.7	14.6	14.7	-16%	-15%	-17%	-6%
	MSEDL	11.9	13.5	11.8	11.4	-30%	-22%	-33%	-27%
	MSPC	12.1	11.5	11.5	11.5	-29%	-33%	-34%	-27%

Therefore, the results indicate that in these P4 problems, the MGPI and MSEDL solutions are too loosely converged for low power cases, but seem reasonably tuned for intermediate power levels. The MGPC and MSPC methods converge nearly-optimally over a broader range of conditions.

4.4.5 Full-core Single State Problem

In this section, the verification of the NOPC-CMFD method continues for the full-core problem–VERA core physics benchmark progression problem 7 (P7). P7 represents Watts Bar Nuclear Unit 1 at Beginning-of-Life (BOL), with hot full power, equilibrium xenon conditions [98], with different fuel enrichments of 2.11% (red), 2.62% (green), and 3.10% (blue), respectively, as shown in Figure 4.15. The feedback of P7 is TCE. More details about this problem can be found in [98, 34].



2.1	2.6 PY20	2.1	2.6 PY20	2.1	2.6 PY20	2.1	3.1 PY12
2.6 PY20	2.1	2.6 PY24	2.1	2.6 PY20	2.1	3.1 PY24	3.1
2.1	2.6 PY24	2.1	2.6 PY20	2.1	2.6 PY16	2.1	3.1 PY8
2.6 PY20	2.1	2.6 PY20	2.1	2.6 PY20	2.1	3.1 PY16	3.1
2.1	2.6 PY20	2.1	2.6 PY20	2.6	2.6 PY24	3.1	
2.6 PY20	2.1	2.6 PY16	2.1	2.6 PY24	3.1	3.1	
2.1	3.1 PY24	2.1	3.1 PY16	3.1	3.1		
3.1 PY12	3.1	3.1 PY8	3.1				

Figure 4.15: Assembly and poison layout of the full-core problem (quarter symmetry). This is regenerated from [98]

.

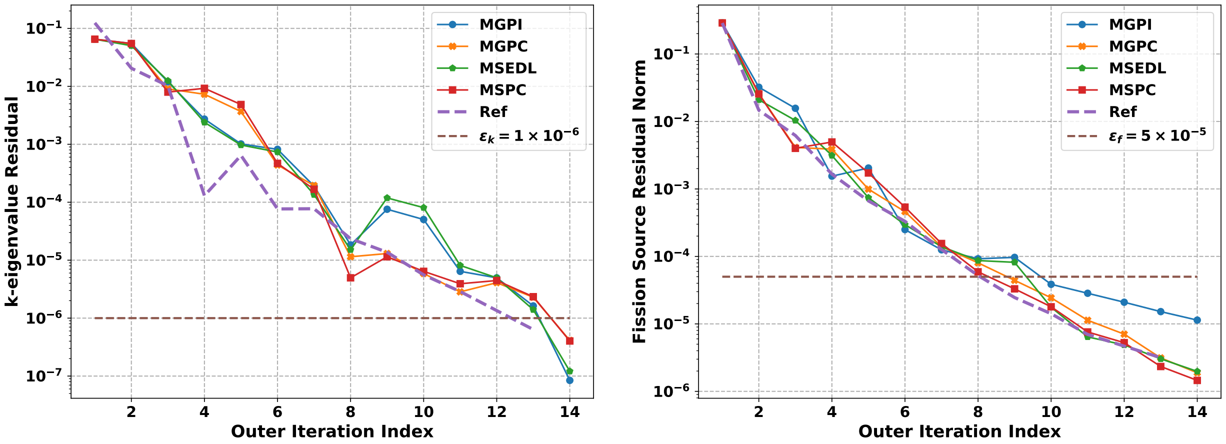
The convergence criteria for this problem are defined by the tolerance for the fission source residual $\epsilon_f = 5 \times 10^{-5}$ and the tolerance of the $k-$eigenvalue residual $\epsilon_f = 10^{-6}$. The reference result of the number of outer iterations for the classical CMFD method in this problem without any feedback is obtained using the MSED solver. The value of the reference result is 13. The convergence properties of the other CMFD solvers for simulating P7 are summarized in Table 4.16. All the simulations of P7 are performed with 464 cores on the Sawtooth high-performance computing system of the Idaho National Laboratory.

It turns out that MPACT with any of the CMFD solvers converges P7 in 14 outer iterations, indicating that the MGPI and MSEDL are specifically tuned for the nominal power case. In fact, the convergence specification for the MSEDL used in this thesis is the same as the convergence specification adopted to mitigate the instability of MSED in P7 presented in [32]. The reason why all the solvers converge the P7 in 14 outer iterations is that the convergence of the simulation is determined by the convergence of k-eigenvalue rather than the convergence of the fission source. As shown in Figures 4.16, the fission source has been converged by 9 or 10 outer iterations, while the k-eigenvalue is converged at the 13^{th} or 14^{th} iteration. In terms of the convergence of the fission source, it can be seen that MPACT with the NOPC-CMFD solvers and the MSEDL solver with relaxation can converge the fission source in a similar behavior to the reference result. The residual of the converged fission source for the NOPC-CMFD methods and the MSEDL method is much smaller than that for the MGPI solver. Therefore, it can be concluded that the NOPC-CMFD has comparable performance to the classical CMFD in the P7 problem.

In terms of efficiency, the run time of MPACT with the MGPC method is reduced by around 12% compared to the run time of MGPI. The MSPC method and MSEDL method have similar run times. Their run time reduction is around 25%.

Table 4.16: Convergence properties of the MPACT with different iteration schemes in P7. The "Ref" results are obtained without feedback using the MSED solver.

Cases	Outer Iterations	Shielding Calculations	Time (min)	Time Difference
Ref	13	-	19.0	-
MGPI	14	7	47.4	-
MGPC	14	6	41.5	-12%
MSEDL	14	6	35.5	-25%
MSPC	14	6	35.3	-26%



(a)

(b)

Figure 4.16: Variation of the $k-$eigenvalue residual and the fission source residual with the outer iteration for P7. (a) shows that the results of k-eigenvalue residual, while (b) shows that the results of the fission source residual. The figure shows that the fission source residual is converged at 9^{th} or 10^{th} outer iteration, while the k-eigenvalue is converged at 13^{th} or 14^{th} outer iteration.

4.4.6 Multistate Full-core Depletion Problem

The most important verification of the NOPC-CMFD method is the full-core cycle depletion problem. The problem used in this section is the VERA core physics benchmark problem 9 (P9). The model is a 32-state depletion problem developed on the basis of the Watts Bar Nuclear Unit 1 Cycle 1. More details about this problem can be found in [4, 98]. No reference result of the performance for classical CMFD is obtained for this problem, so the performance of the different solvers are compared with each other. The results are shown in Table 4.17.

Compared to the MGPI case, the reduction of the number of outer iterations from using the NOPC-CMFD solver is more than 340. The reduction on average is 10 outer iterations per state. Compared to the MSEDL case, the total reduction in the number of outer iterations is more than 130, roughly 4 outer iterations per state.

It is noted that in this problem, MPACT with the MGPI solver performs the fewest number of shielding calculations. The cases using the NOPC-CMFD methods perform around 14 (or more) shielding calculations, which is about 0.44 more shielding calculations per state. More shielding calculations are performed due to the estimation of the feedback intensity. When the fission power is perturbed, the change of the state variables at some states will trigger the shielding calculation to be executed and result in more overall shielding calculations.

In terms of efficiency, it can be seen that the run time reduction for MGPC is 27%. When the multilevel method is used, the run time reduction can be as large as 43%. The MSPC case converges faster than the MSEDL case, indicating that the NOPC-CMFD method can improve the robustness and efficiency of MSED in problems with feedback.

The results presented show the relaxation-free CMFD method–nearly-optimal partially converged CMFD–can improve the efficiency of the current CMFD methods and improve the robustness of the advanced CMFD methods over the PWR operational space.

Table 4.17: Convergence properties of the MPACT with different iteration schemes in P9.

	Outer Iterations	Shielding Calculations	Wall Time (hr)	Time Difference
MGPI	861	169	16.1	-
MGPC	514	182	11.8	-27%
MSEDL	647	190	10.0	-38%
MSPC	507	184	9.1	-43%

4.5 Summary

The work presented in this chapter provided the numerical verification for the NOPC-CMFD method, a robust relaxation-free multiphysics iteration scheme for CMFD-accelerated neutron transport k-eigenvalue problems Chapter 3. Theoretically, the convergence rate of the NOPC-CMFD method in the problem with feedback is similar to that of the fully converged CMFD method in the same problem without feedback. The extensive results presented here essentially verify this theoretical prediction, and demonstrate that the benefits are observable for realistic whole-core PWR cycle depletion calculations. The method, to the best of our knowledge, is the first method that dynamically stabilizes the iteration scheme by adjusting the partial convergence of the low-order diffusion solutions using problem-dependent information.

First, the convergence performance of the CMFD method in MPACT for the fuel assembly problem with feedback, VERA core physics benchmark progression problem 6 (P6), was analyzed. These results showed that:

- the range of the nearly-optimal partial convergence is dependent on the feedback intensity of the problem;
- the partial convergence of the low-order diffusion solution is more important than the relaxation;
- when the low-order diffusion solution is properly partially converged, the number of outer iterations MPACT takes to converge the problem with feedback is the same as the non-feedback case.

These findings agree with the predictions in Chapter 3.

Next, the implementation of the NOPC-CMFD method was described. The key to implementing the method is to estimate the global feedback intensity of the problem. In this chapter, it has been proposed that the pin-wise feedback intensity can be reasonably estimated from a pin-wise perturbation of the flux in a red-black manner. The assumptions made with respect to the feedback in Chapter 3 were verified, and a sensitivity study was performed to determine the parameters used for the estimation of this value. An additional sensitivity study was also performed for deriving the global feedback intensity from a characteristic feedback intensity. The NOPC-CMFD method was implemented based on the CMFD solvers available in MPACT–Multigroup CMFD solver with power iteration (MGPI) solver and the MSED solver. The two new CMFD methods were referred to as the MGPC method and the MSPC method.

Finally, numerical results were presented to show the performance of these two NOPC-CMFD implementations for 3D problems with different scales and forms of feedback. The VERA core physics benchmark progression problem 1 (fuel pin problem), 6 (fuel assembly problem), 4 (3×3 core problem), 7 (full-core single-state problem) and 9 (full-core multi-state depletion problem) with feedback from thermal hydraulics, critical boron search and equilibrium Xenon were used. It was shown that the MGPC and MSPC methods were much more stable and robust than the current CMFD methods of MPACT over a wide range of PWR problems and conditions. For the multi-state depletion problem, compared to the MGPI, the reduction in the number of outer iterations was more than 340 for 32 states. The associated wall time reduction was about 43%. More importantly, it has been shown that the relaxation-free method in problems with feedback has similar performance to the CMFD method in problems without feedback.

The work presented in this chapter is important because it is a relaxation-free method derived based on the iteration scheme used in the neutronics simulations. It is also believed that the NOPC-CMFD method provides a way to improve the stability and robustness of the current CMFD method and CMFD methods that may be proposed in the future.

The main drawback of the NOPC-CMFD method is that it relies on the estimation of the global feedback intensity. This parameter is somewhat complex, and not necessarily straightforward to calculate accurately from realistic simulations. Additionally, the initial guess of the feedback intensity (which should be conservative–but not too conservative) will impact the overall performance (although not as much as the choice of relaxation factor). However, the NOPC-CMFD method should still be considered of practical significance and value because it does not require substantial modification to the iteration schemes that are already widely used by existing high-fidelity whole-core simulators. The required modifications to these existing codes are merely (1) implementation of an iteration dependent Wielandt shift, and (2) a step to estimate γ. Therefore, it may be easily adopted in these codes and guaranteed to improve the stability and robustness of the multiphysics iterations.

However, the issues that surround the calculation of γ are likely to always persist, so we view this work as a temporary solution. For the development of better methods, we propose a different direction where it is more desirable that the method not rely on the estimation of the feedback intensity.

CHAPTER 5

Multiphysics Diffusion Acceleration with X-CMFD

In this chapter, we present the derivation and the implementation of the X-CMFD method, where multiphysics feedback is applied at the power iteration of the low-order eigenvalue solver.

We first formulate, for the same model from Chapter 3, a robust CMFD-based iterative method that (i) requires no relaxation, and (ii) has convergence properties similar to CMFD for problems without feedback. The primary idea underpinning the method is to appropriately and explicitly account for the nonlinear feedback in the low-order problem rather than the partial convergence of the low-order problem. In the continuous problem, the method is referred to as the Nonlinear Fully Coupled Diffusion Acceleration (NFCDA) method. The method is shown to be unconditionally stable, and has the same and even smaller convergence rate in problems with feedback compared to the NDA in the problems without feedback.

The basic assumption of NFCDA is that the low-order diffusion problem is coupled with feedback. However, the solution process of coupling the diffusion and feedback is unstable [100]. The X-CMFD method can be understood as a method that solves the low-order fully coupled diffusion problem robustly, therefore, can have the same convergence rate as the NFCDA when the low-order diffusion is fully converged.

The X-CMFD method is implemented in MPACT and optimized for practical simulations. The implementation is based on the current CMFD solvers that solve the low-order diffusion equation formulated from odCMFD [16] in MPACT. The work presented verifies the assumptions and predictions made in the derivation of the X-CMFD method. And the numerical results are presented to illustrate the performance of the new method in problems of different scales with feedback of various forms.

This chapter proceeds with a description of the neutron transport problem (with model nonlinear feedback), and then describes in detail the NFCDA and the X-CMFD iteration scheme in Section 5.1. The Fourier analysis and results are then provided in Section 5.2. Numerical results from a 1D simple problem are also provided to confirm the predictions of the Fourier analysis and demonstrate the robustness of the X-CMFD method for a wide variation of parameters in the model problem. Next, the X-CMFD algorithm is implemented in MPACT and the details are shown in Section 5.3. The stability of X-CMFD and the predictions made in Fourier analysis are eventually verified in Section 5.4 for some practical problems.

5.1 Motivation and Model Problem

5.1.1 Motivation

The development of a stable iteration scheme irrespective of the convergence of the CMFD solution and problem feedback intensity is the motivation for developing the X-CMFD method. This is the same motivation as Chapter 3 and Chapter 4. However, we have the additional motivation that we wish to construct an iteration scheme that does not rely on the estimation of the problem-dependent feedback. The basic idea of X-CMFD–coupling the low-order problem in the CMFD-accelerated scheme to the feedback from other physics to improve the stability –is similar to that of [101, 34]. However, the previous works did not make any theoretical contribution, or perform a Fourier analysis which is what we do here. Additionally, the overall procedure shown in Figure 2.1 will not be altered. The eventual Fourier analysis results indicate that the method is different from the methods implemented in [101, 34].

5.1.2 Model Problem

The model problem used to describe and analyze the X-CMFD method is a slab-geometry, mono-energetic, isotropically scattering, eigenvalue neutron transport problem with domain size X. The feedback model assumes cross-sections that depend linearly on the local value of the scalar flux. This is the same model problem developed in Chapter 3. The model also imposes a normalization condition to make the eigenfunction unique. Reflective boundary conditions are set on both sides of the problem domain. The transport problem of interest is given in Eqs. (3.1).

$$\mu\frac{\partial\psi\left(x,\mu\right)}{\partial x}+\Sigma_t\left(x,\boldsymbol{S}\right)\psi\left(x,\mu\right)=\frac{1}{2}\Big[\Sigma_s\left(x,\boldsymbol{S}\right)+\lambda\nu\Sigma_f\left(x,\boldsymbol{S}\right)\Big]\phi(x)\,, \tag{3.1a revisited}$$

$$\phi\left(x\right)=\int_{-1}^{1}\psi\left(x,\mu\right)d\mu\,, \tag{3.1b revisited}$$

$$\psi\left(0,\mu\right)=\psi(0,-\mu)\,,\qquad 0<\mu\leq 1\,, \tag{3.1c revisited}$$

$$\psi\left(X,\mu\right)=\psi\left(X,-\mu\right)\,,\qquad -1\leq\mu<0\,, \tag{3.1d revisited}$$

$$\frac{1}{X}\int_0^X\int_{-1}^{1}\psi\left(X,\mu\right)d\mu dx=\Phi_0\,, \tag{3.1e revisited}$$

$$\Sigma_i(x,\boldsymbol{S})=\Sigma_i(x,\phi)=\Sigma_{i0}+\Sigma_{i1}\Big[\phi(x)-\Phi_0\Big]\,,\ i=t,a,s,f,c\,. \tag{3.2 revisited}$$

5.1.3 The NFCDA Iteration

To solve the NTE coupled with feedback robustly, both [34] and the work presented in this chapter couple the low-order diffusion problem and the feedback. However, theoretical proof has not been provided to show that this method is stable for the transport calculation.

In this subsection, we introduce a model problem for the NFCDA method, where the low-order diffusion problem is fully coupled with the feedback.

As will be shown later in this chapter, the NFCDA is a robust method that is as stable in problems with feedback as the NDA is without feedback.

The iteration process of NFCDA in the model problem is shown in Algorithm 11.

Algorithm 11 Fully coupled NFCDA for k-eigenvalue problem with feedback

```
1:  Input maximum outer iteration number $N$; tolerance $\epsilon_c$, $\epsilon_f$; initial guess $\boldsymbol{\phi}^{(0)}$, $J^{(0)}$, $\lambda^{(0)}$.
2:  for $n = 0, 1, \ldots, N$ do
3:      Update the macroscopic cross-section $\Sigma_i^{(n)}$ with the latest flux using Eq. (3.2). Calculate the
        diffusion coefficient and the nonlinear coupling term, $\hat{D}^{(n)}$.
4:      Solve the low-order diffusion eigenvalue problem:

        $-\frac{d}{dx}D^{\left(n+\frac{1}{2}\right)}(x)\frac{d}{dx}\phi^{\left(n+\frac{1}{2}\right)}(x)+\left(\Sigma_a^{\left(n+\frac{1}{2}\right)}(x)+\hat{D}^{(n)}(x)\right)\phi^{\left(n+\frac{1}{2}\right)}(x)$
        $\quad = \lambda^{\left(n+\frac{1}{2}\right)}\nu\Sigma_f^{\left(n+\frac{1}{2}\right)}(x)\,\phi^{\left(n+\frac{1}{2}\right)}(x)\ ,$          (Alg 11.1)

        where

        $\Sigma_i^{\left(n+\frac{1}{2}\right)}(x) = \Sigma_{i0}(x) + \Sigma_{i1}(x)\left(\phi^{\left(n+\frac{1}{2}\right)}(x) - \Phi_0\right) \quad i = t, s, c, \gamma, f \cdots .$          (Alg 11.2)

5:      Perform a transport sweep:

        $\mu\left[\frac{\partial}{\partial x} + \Sigma_t^{(n)}(x)\right]\psi^{(n+1)}(x,\mu) = \frac{1}{2}\left[\Sigma_s^{(n)}(x) + \lambda^{\left(n+\frac{1}{2}\right)}\nu\Sigma_f^{(n)}(x)\right]\phi^{\left(n+\frac{1}{2}\right)}(x)\ .$
                                                                                  (Alg 11.3)

        Note here the cross sections on the fine mesh are from step 3.
6:      Calculate the scalar flux $\phi^{(n+1)}$ and current $J^{(n+1)}$. Let $k_{eff}^{(n+1)} \leftarrow 1/\lambda^{(n+1/2)}$.
7:      if $\left|k_{eff}^{(n+1)} - k_{eff}^{(n)}\right| < \epsilon_c$ and $\left\|\mathcal{F}^{(n)}\phi^{(n)} - \mathcal{F}^{(n+1)}\phi^{(n+1)}\right\| < \epsilon_f \left\|\mathcal{F}^{(n)}\phi^{(n)}\right\|$ then
8:          break
9:      end if
10: end for
```

It should be noted that Eqs. (Alg 11.1) is assumed to be solved exactly without any stability issues. However, instability in solving the diffusion problem coupled with feedback from other physics is well-known [100]. Therefore, the application of NFCDA should be limited by the stability of the iterative method used to obtain the solution of the low-order diffusion problem.

5.1.4 Iteration Procedure of X-CMFD

The X-CMFD algorithm for the continuous neutron transport eigenvalue problem with nonlinear feedback is detailed in Algorithm 12. In this algorithm, the steps provided are for the $(n+1)^{th}$ iteration of the scheme. Prior to starting the first iteration, user-provided initial guesses are necessary for $\phi^{(0)}(x)$, $J^{(0)}(x)$, and $\lambda^{(0)}$. The essential new feature of the X-CMFD method occurs in step 6, where the low-order problem is iterated to appropriately converge the flux-dependent cross sections. It should be noted that the $\hat{D}$ term is only updated once per outer iteration.

Algorithm 12 X-CMFD for k-eigenvalue problem with feedback

```
 1: Input maximum outer iteration number $N$, inner coupling iteration number $M$; tolerance $\epsilon_c$,
    $\epsilon_f$; initial guess $\boldsymbol{\phi}^{(0)}$, $J^{(0)}$, $\lambda^{(0)}$.
 2: for $n = 0, 1, \ldots, N$ do
 3:     Update the macroscopic cross-section $\Sigma_i^{(n)}$ with the latest flux using Eqs. (3.2). Calculate
        the diffusion coefficient and the nonlinear coupling term, $\hat{D}^{(n)}$.
 4:     for $m = 0, 2, \ldots M-1$ do
 5:         Update the cross section with the latest flux $\phi^{(n,m)}$, and update the diffusion coefficient.
 6:         Solve the low-order diffusion eigenvalue problem:
            $$-\frac{d}{dx}D^{(n,m)}(x)\frac{d}{dx}\phi^{(n,m+1)}(x) + \left(\Sigma_a^{(n,m)}(x) + \hat{D}^{(n)}(x)\right)\phi^{(n,m+1)}(x) = \lambda^{(n,m+1)}\nu\Sigma_f^{(n,m)}(x)\,\phi^{(n,m+1)}(x)\ , \quad \text{(Alg 12.1)}$$
            with $L$ power iteration using WS ratio $r$.
 7:     end for
 8:     Let $\lambda^{\left(n+\frac{1}{2}\right)} \leftarrow \lambda^{(n,M)}$ and $\phi^{\left(n+\frac{1}{2}\right)} \leftarrow \phi^{(n,M)}$. Then perform transport sweep:
        $$\mu\Big[\frac{\partial}{\partial x} + \Sigma_t^{(n)}(x)\Big]\psi^{(n+1)}(x,\mu) = \frac{1}{2}\left[\Sigma_s^{(n)}(x) + \lambda^{\left(n+\frac{1}{2}\right)}\nu\Sigma_f^{(n)}(x)\right]\phi^{\left(n+\frac{1}{2}\right)}(x)\ . \quad \text{(Alg 12.2)}$$
        Note here the cross sections on the fine mesh are from step 3.
 9:     Calculate the scalar flux $\phi^{(n+1)}$ and current $J^{(n+1)}$. Let $k_{eff}^{(n+1)} \leftarrow 1/\lambda^{(n+1/2)}$.
10:     if $\left|k_{eff}^{(n+1)} - k_{eff}^{(n)}\right| < \epsilon_c$ and $\left\|\mathcal{F}^{(n)}\phi^{(n)} - \mathcal{F}^{(n+1)}\phi^{(n+1)}\right\| < \epsilon_f \left\|\mathcal{F}^{(n+1)}\phi^{(n+1)}\right\|$ then
11:         break
12:     end if
13: end for
```

5.2 Fourier Analysis

The details of the Fourier analysis of the model problems are shown in Appendix A. The details are omitted here for brevity.

5.2.1 Expression for Spectral Radius

For the continuous version, the predicted spectral radius is generally representative of any fine-mesh discretization. For the discretized problem, there are two spatial meshes–a fine mesh for the transport sweep and a coarse mesh for the low-order diffusion eigenvalue calculation, with q fine cells per coarse cell. It is assumed that both meshes are uniform with mesh cell thickness h for the fine mesh and $\Delta = qh$ for the coarse mesh.

The main Fourier analysis result is the iteration eigenvalue θ, or growth factor for the error. The derivation of the full result is straightforward but algebraically intensive, and we omit the details here for brevity, it can be found in Appendix A. The derivation procedure amounts to a linearization of the X-CMFD algorithm given in the previous section, insertion of a Fourier Ansatz, and manipulation of the resulting algebraic equations to obtain the iteration eigenvalue as a function of Fourier frequency.

The spectral radius is then found by maximizing θ over all possible Fourier modes:

$$\rho = \max_{\omega} \left|\theta(\omega)\right| . \tag{5.2}$$

5.2.1.1 NFCDA

The iterative eigenvalue for continuous the NFCDA is

$$\theta(\omega) = \frac{f_{NDA}\left(\omega\right)}{b\left(\omega\right)} - \gamma f_{TS}\left(\omega\right) . \tag{5.3}$$

Here ω, f_{NDA} and f_{TS} have the same definitions as those in Section 3.2. They are shown again in Eqs. (5.4) along with $a\left(\omega\right)$ and $b\left(\omega\right)$.

5.2.1.2 X-CMFD

The Fourier analysis result for X-CMFD (Algorithm 12) is:

$$\theta(\omega) = \begin{cases} \Big(a(\omega) + \frac{1-a(\omega)}{b(\omega)} - \gamma\Big) f_{TS}\left(\omega\right) - \frac{3}{\omega^2}\frac{1-a(\omega)}{b(\omega)}\Big[1 - f_{TS}\left(\omega\right)\Big] , & \text{continuous problem} , \\ \max\Big|eig\Big(\mathbf{T}\Big(\omega\Big)\Big)\Big| , & \text{discretized problem} , \end{cases} \tag{5.4a}$$

where:

$$\mathbf{T}(\omega) = \tilde{\mathbf{H}}(\omega)\Big(1-\gamma\Big) - \Big[1-a(\omega)\Big]\mathbf{1}\frac{3\Sigma_t\Delta(e^{i\Sigma_t\Delta\omega}-1)\tilde{\mathbf{G}} + \gamma 3(\Sigma_t\Delta)^2\frac{\mathbf{1}^T}{q}\tilde{\mathbf{H}}}{\gamma 3\Big(\Sigma_t\Delta\Big)^2 + 2 - 2cos\Big(\Sigma_t\Delta\omega\Big)}\,, \tag{5.4b}$$

$$f_{TS}\left(\omega\right) = \frac{\arctan\left(\omega\right)}{\omega}\ , \tag{3.20a revisited}$$

$$a(\omega) = \Big[\left(\Lambda^L(\omega)-1\right)\frac{\gamma}{g(\omega)} + \Lambda^L(\omega)\Big]^M\,, \tag{5.4c}$$

$$b(\omega) = \frac{\gamma}{g(\omega)} + 1\,, \tag{5.4d}$$

$$\tilde{c} = 1 - \left(1-r\right)\left(1-c\right), \tag{3.20c revisited}$$

$$g(\omega) = \begin{cases} \frac{1}{3}\omega^2\,, & \text{continuous problem}\,, \\ \frac{2-2cos\Big(\Sigma_t\Delta\omega\Big)}{3\Big(\Sigma_t\Delta\Big)^2}\,, & \text{discretized problem}\,, \end{cases} \tag{5.4e}$$

$$\Lambda(\omega) = \frac{1-\tilde{c}}{1-\tilde{c}+g(\omega)}\,. \tag{5.4f}$$

In Eqs. (5.4), ω is the Fourier frequency and is a multiple of $\frac{\pi}{\Sigma_t X}$. $\tilde{\mathbf{H}}$ is the error transition matrix for the fine mesh scalar flux due to the discretized transport sweep operator. $\tilde{\mathbf{G}}$ is the operator for the coarse mesh surface net current obtained from the fine mesh surface angular flux. The definition of these two terms can be found in [16, 94].

We have used a WS power iteration to solve Eq. (Alg 12.1), and thus we incorporate typical solver parameters such as the shift value and the number of power iterations as parameters in the Fourier analysis. Therefore, this Fourier analysis includes the factors L and r that apply to the iterative method used to solve Eq. (Alg 12.1).

For X-CMFD, the nonlinear term is calculated once after every transport sweep. If the nonlinear coupling term is calculated together with the diffusion term in Algorithm 12 by

$$\hat{D}^{(n,m)}\left(x\right) = \frac{\frac{d}{dx}\left[J^n\left(x\right) + D^{n+\frac{1}{2},m}\left(x\right)\frac{d}{dx}\phi^{\left(n+\frac{1}{2},m\right)}\left(x\right)\right]}{\phi^{\left(n+\frac{1}{2},m\right)}\left(x\right)}\,, \tag{5.5}$$

and no feedback is present, the expression for the growth factor is:

$$\theta\left(\omega\right) = f_{TS}\left(\omega\right) - \frac{3M}{\omega^2}\Big[1 - f_{TS}\left(\omega\right)\Big]\,. \tag{5.6}$$

We will discuss the implications of updating $\hat{D}$ with the inner cross section update later in Section 5.2.2.3, but this is generally not a good idea.

5.2.2 Analysis Results

5.2.2.1 NFCDA

The spectral radius of the NFCDA method is shown in Figure 5.1. The feedback intensities used are of different orders of magnitude. It could be seen that the method is stable for a very large range in the problem with feedback of different intensities. And when feedback is very strong $\gamma = 1e-1$, the NFCDA converges faster than the NDA in problem without feedback.

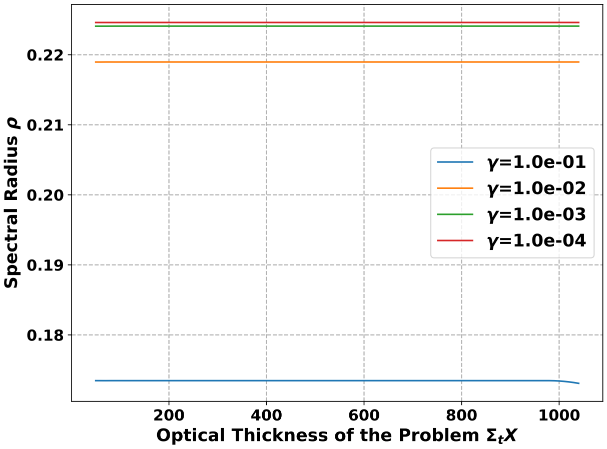


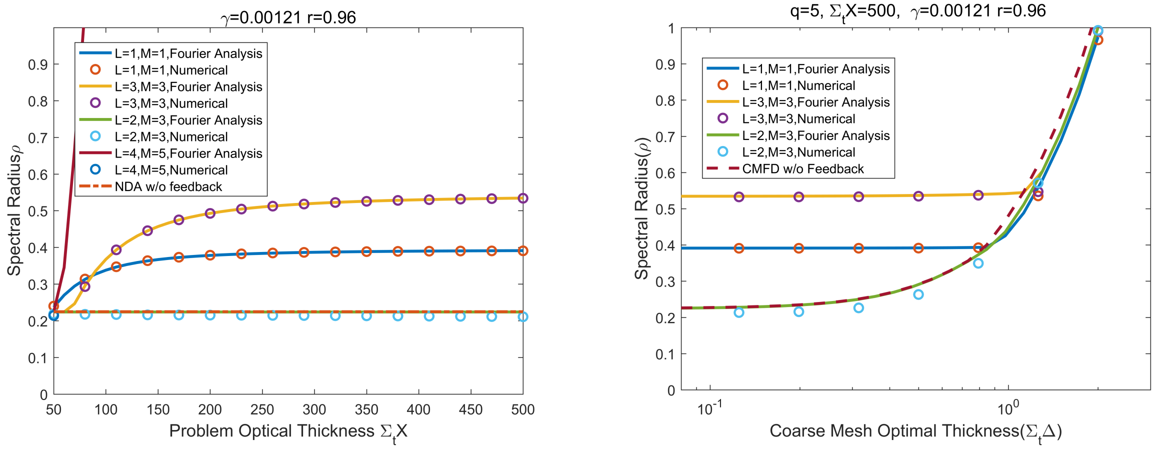
Figure 5.1: Stability of the NDA with fully-coupled low-order problem.

The result is very important, *because it indicates that if the low-order diffusion problem is fully coupled with feedback and can be converged without instability, the NFCDA should have almost identical or even better performance compared to the NDA in problems without feedback.*

5.2.2.2 Validation of Fourier Analysis Results

In Figures 5.2, theoretical spectral radius predictions obtained from the Fourier analysis and numerically-obtained spectral radius estimates are compared for both continuous and discretized equations to validate the Fourier analysis results of X-CMFD. The NFCDA could not be validated, because it is a hypothetical method based on the fully-coupled low-order diffusion problem that is solved robustly. How the low-order diffusion problem is solved is explicitly undefined.

We observe that the Fourier analysis predictions agree well with the numerical results obtained from a test code. Moreover, plots in Figure 5.2a illustrate that the spectral radius is affected by the size of the problem domain, i.e. the larger the problem size, the less stable the iteration scheme. However, it can be observed that when $M = 3$ and $L = 2$, this dependence on problem size disappears, and the spectral radii for problems with optical thickness up to 500 mean free paths are almost constant and around 0.2247 (the optimal convergence rate for NDA in the non-feedback problem!). With this combination of M and L, X-CMFD has a nearly identical asymptotic convergence behavior compared to CMFD in the non-feedback discretized problem, which can be found in Figure 5.2b.



(a)

(b)

Figure 5.2: Comparison of Fourier analysis and numerical results of X-CMFD for some example feedback problems. (a) shows the results of the continuous problem. And (b) show the results of the discretized problem.

5.2.2.3 Nonlinear Coupling Term

Now we explain why the nonlinear coupling term in Algorithm 12 must be only calculated once at each outer iteration. Eq. (5.6) is rewritten as:

$$\theta\left(\omega\right) = f_{TS}\left(\omega\right) - \frac{3M}{\omega^2}\left(1 - f_{TS}\left(\omega\right)\right) = f_{NDA}\Big[1 - f_{TS}\left(\omega\right)\Big]\omega^2\Big[1 - f_{TS}\left(\omega\right)\Big] \,. \tag{5.7}$$

Since

$$\lim_{\omega\to 0}\frac{3}{\omega^2}\left(1 - f_{TS}\left(\omega\right)\right) = 1\,, \tag{5.8a}$$

$$\lim_{\omega\to 0} f_{NDA}\left(\omega\right) = 0\,, \tag{5.8b}$$

therefore

$$\lim_{\omega\to 0}\theta\left(\omega\right) = M - 1\,. \tag{5.9}$$

For $M \geq 2$,

$$\rho \geq \left|\theta\left(\omega\right)\right| \approx M - 1\,. \tag{5.10}$$

The results indicate that when the nonlinear coupling term is calculated every inner iteration, the iteration scheme will become unstable even for the problems without feedback. The explanation is that the nonlinear coupling term is used as a correction to the Fick's law approximation of the neutron current, and in particular that correction should be independent of the diffusion coefficient (assuming it is correctly defined for each condition of the system). Therefore, only the terms from the transport sweep should be used. Consequently, the nonlinear coupling term $\hat{D}$ should only be calculated once per outer iteration.

5.2.2.4 Fully Converged Low-order Problem

In Chapter 3, it has been shown that if the low-order problem is fully converged, the Picard iteration scheme used in MPACT becomes unstable. It is worthwhile to see whether using the coupling between the low-order problem and the feedback will stabilize the calculation, so that when the low-order problem is fully converged, the convergence becomes better. Here we assume that $L = \infty$. Therefore,

$$a\left(\omega\right) = \left(-\frac{\gamma}{g\left(\omega\right)}\right)^{M}. \tag{5.11}$$

The plot of the spectral radius for the continuous problem is shown in Figure 5.3. The results show

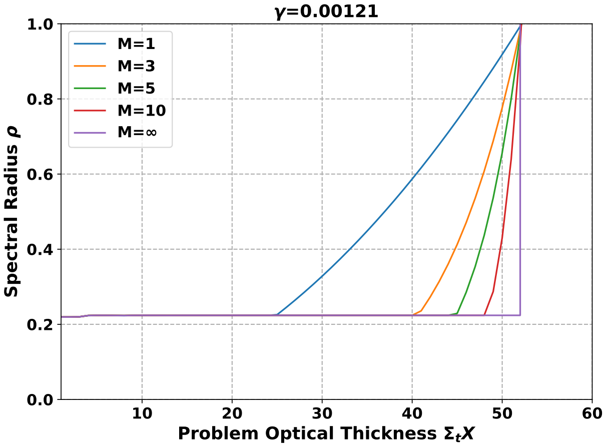


Figure 5.3: Spectral radius as a function of the problem optical thickness for different M. For each inner iteration, the diffusion problem is converged first then the feedback is applied.

that if the low-order problem is fully converged, using inner coupling will only reduce the spectral radius for the problem in which the Picard iteration scheme is already stable. If the Picard iteration scheme is unstable, i.e. $\Sigma_t X > 52$, the inner coupling will not be stable.

5.2.3 X-CMFD Iteration Scheme Design

The results from the previous subsection illustrate that the optimal spectral radius of 0.2247 can be achieved with some combination of power iteration number L and inner iteration number M in X-CMFD. Here we attempt to find whether there exists an optimal set of iteration scheme parameters for X-CMFD that give similar stability behavior as NDA (CMFD) in the non-feedback problem over a wide range of conditions.

5.2.3.1 Scheme Design

Problem parameters representative of a nuclear reactor at operating conditions are used in this part. The problem size is set to be 500 mfp, to allow flatter Fourier modes and make the problem more difficult.

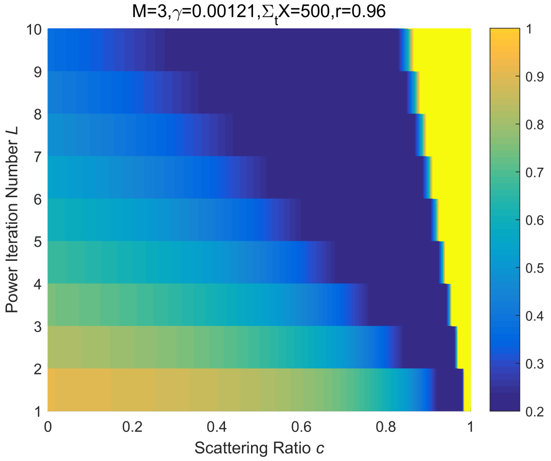
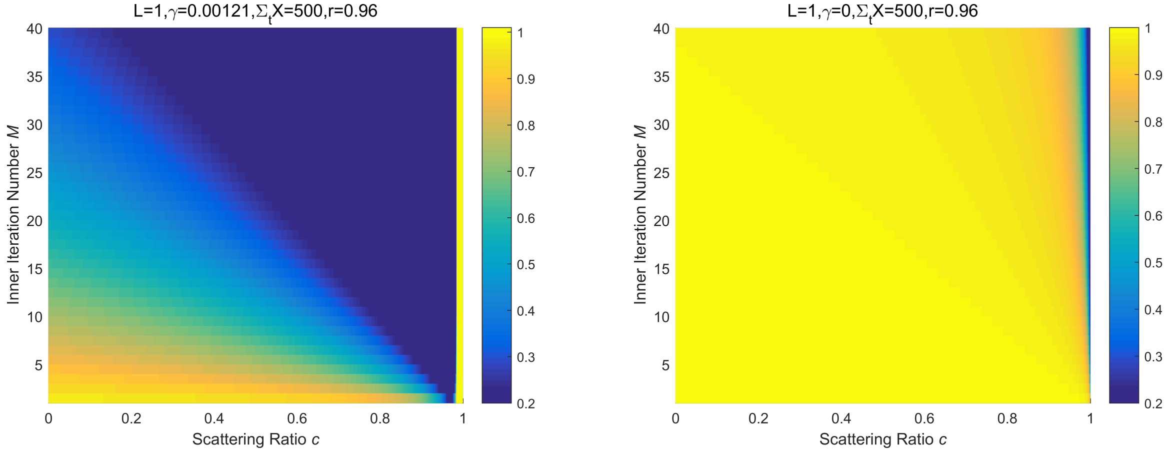


Figure 5.4: The plot shows the effect of the scattering ratio and power iteration number on the X-CMFD scheme analyzed in this chapter.

Figure 5.4 shows that for the chosen set of problem parameters, the most stable option for the L parameter in X-CMFD is $L = 1$, as this value avoids divergence at large scattering ratios. This result is important because it suggests that the work performed in solving the low-order equation should be minimized. It also suggests that *for more general problems, fixed-point iteration schemes with NDA or CMFD acceleration should apply a suitable degree of multiphysics feedback at each power iteration of the diffusion eigenvalue solver.* This is consistent with the concept of "why waste effort converging the low-order problem with the wrong coefficients?". Also, an aggressive Wielandt shift parameter should be used to accelerate the convergence of the coupled power iterations. This is desired because the aggressive Wielandt shift has already been used to accelerate the convergence of the low-order problem [17, 32, 6].



(a) **(b)**

Figure 5.5: Spectral radius as a function of scattering ratio and the number of inner iterations for different γ. (a) shows the results with feedback at nominal PWR conditions. (b) shows the results without feedback.

Figure 5.5a shows how the spectral radius of X-CMFD varies with the scattering ratio c and the number of inner iterations M with $L = 1$. When the system is scattering dominated ($c > 0.9$), the spectral radius is about 0.2247 with a small M. This encouraging result shows that for the problem under consideration, the X-CMFD method converges at the same rate for problems with feedback as the NDA (CMFD) accelerated scheme does for problems without feedback. However, if the scattering ratio is very large ($c > 0.98$), X-CMFD becomes unstable. This problem can be mitigated by reducing the aggressiveness of the Wielandt shift. Additionally,

$$\gamma = \left(\frac{\Sigma_{a,1}}{\Sigma_{a,0}} - \frac{\Sigma_{f,1}}{\Sigma_{f,0}}\right)\frac{\Sigma_{a,0}}{\Sigma_{t,0}}\Phi_0 = \left(\frac{\Sigma_{a,1}}{\Sigma_{a,0}} - \frac{\Sigma_{f,1}}{\Sigma_{f,0}}\right)\Phi_0(1-c)\,, \qquad \text{(3.44 revisited)}$$

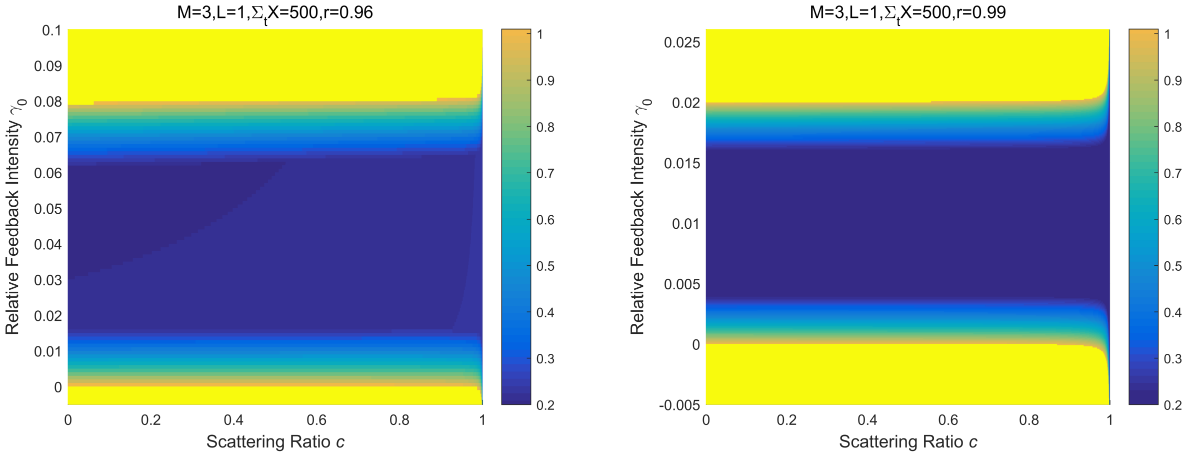
shows that γ is dependent on the scattering ratio. Therefore, for very high scattering ratio materials, γ tends to be small, reducing the instability of the iteration scheme.

In practical simulations, the NDA (CMFD) diffusion equation is only partially converged. Therefore, as suggested by Figure 5.5b, the optimal convergence rate of $\rho = 0.2247$ for standard NDA (CMFD) is not achieved for large problems, even without feedback. By comparing Figure 5.5a and Figure 5.5b, it is found that *X-CMFD makes use of the feedback to become more stable*. This result indicates that in reality, the simulation with feedback may take fewer transport steps to converge than the simulation without feedback for the (otherwise) same problem.

It should be noted that when L is 1, the solution process of the inner coupling turns out to be the power iteration solution of a low-order diffusion problem coupled with feedback. A more encouraging result is that the spectral radius is monotonically decreasing and converges to 0.2247

with inner iteration number M increasing. The observations made here can be interpreted as that the more tightly the X-CMFD diffusion equation is converged, the faster the iteration scheme converges. This result indicates that X-CMFD is robust, and it is possible for users to control the convergence of lower-order diffusion problems without loss of stability. This is not the case for the Picard iteration scheme, in which extra effort used to fully converge the low-order problem can lead to instabilities as suggested by results in Chapter 3.

5.2.3.2 Effect of Feedback Intensity and Wielandt Shift



(a) **(b)**

Figure 5.6: Spectral radius as a function of scattering ratio and the number of inner iterations for different WS ratios. (a) show the results for $r = 0.96$, i.e. the Wielandt shift is relatively mild. (a) show the results for $r = 0.99$, i.e. the Wielandt shift is relatively aggressive.

The effect of the feedback intensity and Wielandt shift on the stability of the X-CMFD scheme is investigated in this section. The relative feedback intensity γ_0 is used to separate the effects of feedback from the scattering ratio. γ_0 is defined in

$$\gamma = \left(\frac{\Sigma_{a,1}}{\Sigma_{a,0}} - \frac{\Sigma_{f,1}}{\Sigma_{f,0}} \right) \Phi_0(1-c) = \gamma_0(1-c)\,, \tag{5.12}$$

to separate the scattering ratio from the feedback intensity. Figures 5.6 show that the X-CMFD method is unstable when there is minor positive feedback ($\gamma_0 < 0$). This indicates that the method is only applicable for reactor configurations with negative reactivity coefficients. The figure also shows that the scheme may be unstable for problems with exceedingly strong feedback. However, Figure 5.6a shows that a less aggressive shift gives the scheme a larger parameter space where it is stable, indicating that reducing the aggressiveness of the Wielandt shift can help to make the overall iteration more stable.

5.2.4 X-CMFD vs NFCDA

The key lesson from the work in this section is that to stabilize the CMFD-accelerated scheme in problem with feedback, *all the relevant physics should be included implicitly in the low-order diffusion equation*. And the feedback should be applied at the power-iteration-level of low-order eigenvalue solver, i.e. L should be 1. The $\hat{D}$ term should only after the transport sweep and not in the inner iteration loop.

For the region where the X-CMFD is stable,

$$\begin{aligned}\lim_{M\to\infty}\theta\left(\omega\right) &= \lim_{M\to\infty}\left(a\left(\omega\right)+\frac{1-a\left(\omega\right)}{b\left(\omega\right)}-\gamma\right)f_{TS}\left(\omega\right)-\frac{3}{\omega^2}\frac{1-a\left(\omega\right)}{b\left(\omega\right)}\Big[1-f_{TS}\left(\omega\right)\Big]\,, \\ &= \frac{f_{NDA}\left(\omega\right)}{b\left(\omega\right)}-\gamma f_{TS}\left(\omega\right)\,.\end{aligned} \tag{5.13}$$

The expression is the same as Eq. (5.3), that is the growth factor of the NFCDA-accelerated scheme assuming that the low-order diffusion is fully coupled with feedback with other physics, and solved exactly.

Therefore, the X-CMFD for the continuous problem can be interpreted as a variant of NFCDA that uses the WS PI method to solve the low-order diffusion problem coupled with feedback robustly. Since NFCDA can have similar performance compared to NDA when the solution of the low-order problem is robust, the X-CMFD becomes a robust method that has comparable performance to the NDA.

It should be noted that the conclusion is made in terms of the solution of the low-order problem, regardless of the specific type of the CMFD method. Therefore, either the NFCDA method or X-CMFD method is a generalized method that addresses the stability issues confronted by NDA and CMFD methods in problems with feedback. We say NFCDA is general because all the methods that have been developed to couple the diffusion problem with feedback can be applied to solve the fully coupled low-order diffusion problem. We say X-CMFD is general because all the modifications that have been developed to improve the performance of the CMFD in problems without feedback [16, 80, 14, 18] are also applicable to the X-CMFD method.

5.3 Numerical Implementation

In this section, we describe how the X-CMFD method is implemented in MPACT. The flowchart of the X-CMFD iteration scheme is shown in Figure 5.7. Compared to the flowchart shown in Figure 2.1, the main modification is that "Solve CMFD Equation" becomes the "Solve X-CMFD Equation".

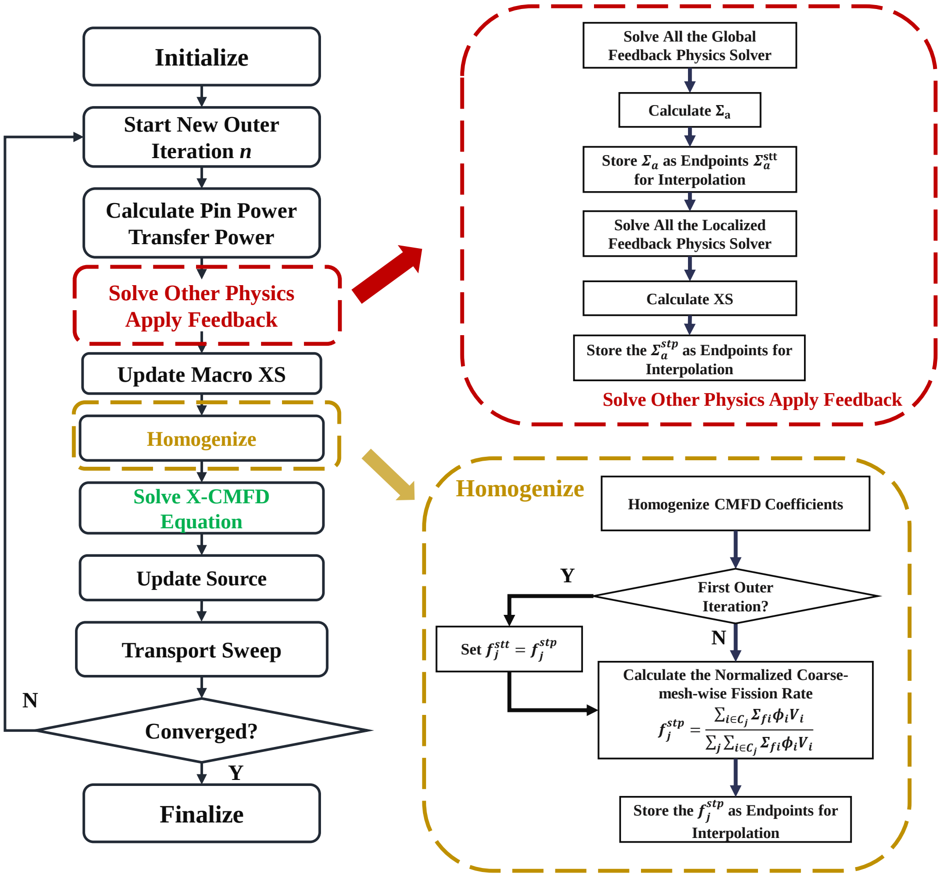

Initialize
Start New Outer Iteration n
Calculate Pin Power Transfer Power
Solve Other Physics Apply Feedback
Update Macro XS
Homogenize
Solve X-CMFD Equation
Update Source
Transport Sweep
Converged?
N
Y
Finalize
Solve All the Global Feedback Physics Solver
Calculate Σ_a
Store Σ_a as Endpoints Σ_a^{stt} for Interpolation
Solve All the Localized Feedback Physics Solver
Calculate XS
Store the Σ_a^{stp} as Endpoints for Interpolation
Solve Other Physics Apply Feedback
Homogenize
Homogenize CMFD Coefficients
First Outer Iteration?
Y
N
Set $f_j^{stt} = f_j^{stp}$
Calculate the Normalized Coarse-mesh-wise Fission Rate
$f_j^{stp} = \frac{\sum_{i \in C_j} \Sigma_{fi} \phi_i V_i}{\sum_j \sum_{i \in C_j} \Sigma_{fi} \phi_i V_i}$
Store the f_j^{stp} as Endpoints for Interpolation

Figure 5.7: Flowchart of X-CMFD iteration scheme in MPACT. For prototype X-CMFD, the only modified part is “Solve X-CMFD Equation”. For the optimized implementation, modifications are made in solution of other physics variables and applying the feedback, and homogenizing the transport cross sections to obtain the CMFD coefficients.

5.3.1 Prototype Implementation

Based on the theoretical formulation presented in Section 5.2, the diffusion problem is solved in the manner illustrated by Figure 5.8a.

In this approach, the other physics equations are solved and the cross sections are updated every power iteration. Consequently, this naïve scheme shown in Figure 5.8a is quite computationally expensive, and it is formidable to use this scheme in a large-scale simulation.

5.3.2 Optimized Implementation

To implement an efficient X-CMFD method, the scheme should be optimized by introducing approximate feedback models to replace the potentially expensive procedure of solving the other physics. As we will show in our numerical results, the X-CMFD scheme can be effectively optimized by simply linearly interpolating the absorption cross section. The idea of interpolating

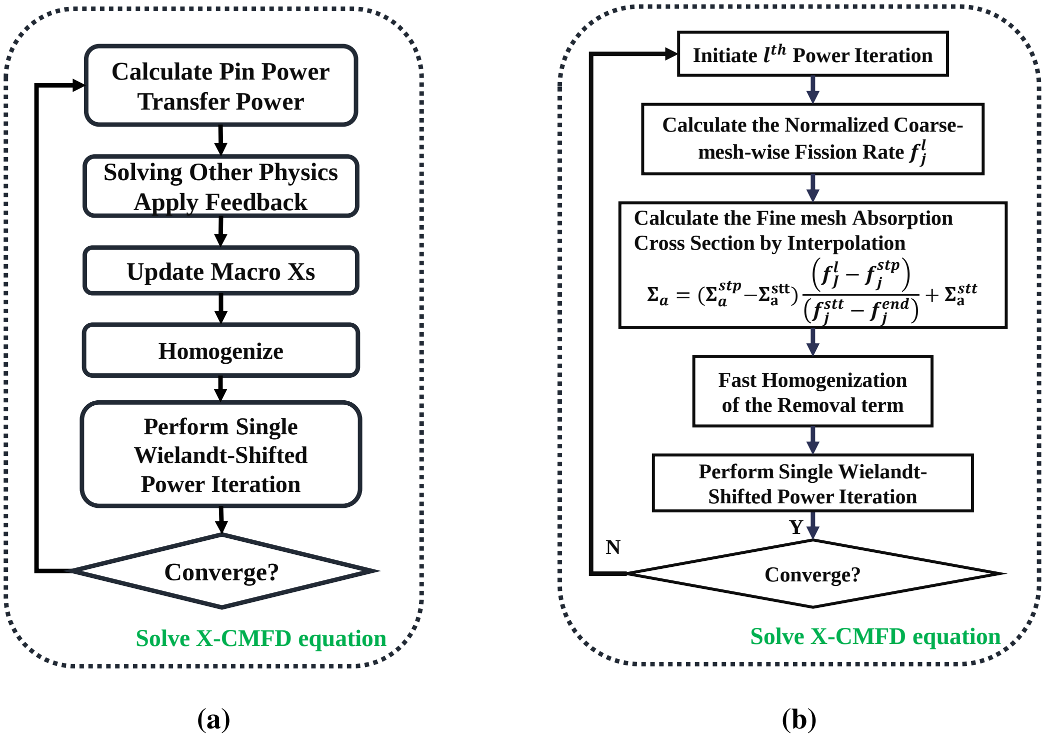

Calculate Pin Power
Transfer Power
Solving Other Physics
Apply Feedback
Update Macro Xs
Homogenize
Perform Single
Wielandt-Shifted
Power Iteration
Converge?
Solve X-CMFD equation
(a)
Initiate l^{th} Power Iteration
Calculate the Normalized Coarse-mesh-wise Fission Rate f_j^l
Calculate the Fine mesh Absorption Cross Section by Interpolation
$\Sigma_a = (\Sigma_a^{stp} - \Sigma_a^{stt})\frac{(f_J^l - f_j^{stp})}{(f_j^{stt} - f_j^{end})} + \Sigma_a^{stt}$
Fast Homogenization of the Removal term
Perform Single Wielandt-Shifted Power Iteration
Y
N
Converge?
Solve X-CMFD equation
(b)

Figure 5.8: Implementation for the solution of X-CMFD equation. (a) shows the implementation of the prototype X-CMFD. (b) shows the optimized implementation of the X-CMFD.

the absorption cross section is drawn from previous work to optimize the application of JFNK in [34, 20], where only the absorption cross section is used from the Jacobian. The optimized X-CMFD accelerated transport scheme is shown in Figure 5.8a with the X-CMFD equation solved following the description in Figure 5.8b.

Prior to the start of the power iteration, the fine mesh absorption cross section and coarse mesh fission rate are stored as the endpoints for interpolation. During the power iteration, the coarse mesh fission rate f^l is calculated first, then the absorption cross section on the fine mesh is interpolated using the fission rate. The endpoints are retrievable after the first two transport iteration, therefore the X-CMFD calculation is performed from the third outer iteration.

The flowchart of obtaining the endpoints is also shown in Figures 5.8. Several implementation details are highlighted here:

- When obtaining the endpoints for the absorption cross section interpolation, the state variables are categorized into two groups: non-local (global) state variables and local state variables. This categorization is made based on whether the feedback makes a local change to the cross sections due to a local change in the flux or a non-local change. The treatment on the feedback is the same as that in the NOPC-CMFD method. More details can be found in Section 4.2.1.

- Though memory is required for storing these coefficients, the additional memory cost is negligible compared to the memory used to store the scattering cross section.
- For the fast homogenization of the removal term of the CMFD coefficients, the removal term can be calculated by subtracting the within-group scattering term from the homogenized transport cross section, since the scattering term is not updated, as suggested by

$$\Sigma_{rm,g} = \Sigma_{tr,g} - \Sigma_{s,g\to g}\,. \tag{5.14}$$

- For the multilevel CMFD method MSED, the one group removal term can also be calculated in the same way. For each one-group power iteration, the removal terms are homogenized with the fine mesh multigroup absorption cross section. The coupling between the low-order solver and the feedback is on the level of the power iteration for the solution of the grey problem.
- The ratio $|f^l - f^{stt}|/|f^{stp} - f^{stt}|$ is limited to not be larger than 5, and the interpolation is not performed in the region where the fission rate is not zero. The summation of f is unity.
- The idea of X-CMFD is different from the low-order coupling in [34], where the iterations are performed in the way where solving the CMFD equations is separated from the solution of the other physics problem. For the X-CMFD method, it is assumed that the low-order diffusion problem is a problem that is fully coupled with the other physics. Therefore, the feedback is applied after every power iteration.

In MPACT, the X-CMFD method has been implemented based on the default MGPI solver and the MSED solver, both of which solve the diffusion equation derived by odCMFD [16]. In this chapter, the new two CMFD solvers are denoted as “MGPLC” and “MSPLC”, where “PLC” is an acronym for the power-iteration-level-coupling, which is the distinguishing feature of the X-CMFD method.

5.4 Numerical Results for Realistic Problems

In this section, a series of coupled problems with increasing complexity starting from a 3D-pin-cell problem to the 3×3 core problem are used to verify the stability of the X-CMFD approach. All the problems are developed from the VERA core physics progression problems [98]. The convergence criteria for the eigenvalue residual and fission source residual are both 10^{-6}. The performance of the X-CMFD solvers is compared with that of the MGPI solver, and the NOPC-CMFD solvers, MGPC and MSPC. Unless otherwise specified, $\lambda_s = 2/3$ in the linear system of MGPI.

5.4.1 Verification of Prototype

The verification of "X-CMFD" in MPACT starts with verifying the prototype implementation where the feedback equations are explicitly solved every outer iteration. The prototype is only verified on the 3D pin-cell problem with feedback from the TH problem. The prototype method is implemented based on MGPI. Table 5.1 compares the number of outer iterations to converge for MPACT using the MGPI solver and the X-CMFD method. No relaxation is applied for both methods just to show that X-CMFD stabilizes the iteration. Note that the low-order problem has been fully converged. It can be seen that if no relaxation is used, and the CMFD diffusion equation is converged tightly, the MGPI cases diverge when the power is higher than 80% of the rated power. The X-CMFD case takes almost a constant number of outer iterations to converge, indicating that the X-CMFD stabilizes the iteration scheme and its stability is insensitive to the feedback intensity which varies with power level. When no feedback is present, the number of outer iterations to converge the P1 case is **16**. Therefore, the X-CMFD method does have a similar performance in 3D P1 problem with feedback to the CMFD method in 3D P1 problem without feedback. However, in terms of run time the prototype X-CMFD method is inefficient because the TH solver is invoked every power iteration on the CMFD equations. The X-CMFD takes more than 15 minutes to converge all the cases while the MGPI takes around 5 minutes to converge the cases with power smaller than 80% rated power. Therefore, this corroborates our assertion that the X-CMFD must be optimized to use an approximate representation of the physics.

Table 5.1: Number of outer iterations to converge 3D P1 with feedback from TH. **div** stands for divergence.

Power Level (%)	20	40	60	80	100	120	140
MGPI	16	16	29	div	div	div	div
X-CMFD (prototype)	16	16	16	16	16	16	17

5.4.2 Single Assembly Problem

The assessment of the X-CMFD method continues with the VERA core physics benchmark progression problem 6 (P6), a single assembly multiphysics problem. The feedback from TH; TH and critical boron search (TC); and TH, critical boron search, and equilibrium xenon (TCE) are taken into account. Table 5.2 shows how the WS ratio affects the stability of the X-CMFD method. It can be seen that, with an increasing WS ratio, the number of outer iterations to converge the P6 cases decreases from 13 to 12. The **12** is the number of iterations MPACT takes to converge the P6 problem without feedback using a tightly converged CMFD solver. The results verify what has been predicted by the X-CMFD Fourier analysis that the tight convergence of the CMFD solution will make the scheme converge faster and the X-CMFD method can converge the problem with feedback at a constant rate without concern over the form or intensity of feedback.

Table 5.2: Number of outer iterations to converge P6 with 140% rated power

CMFD Solver	Feedback	Wielandt Shift Ratio								
		0.90	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98
MGPLC	TH	12	12	12	12	12	12	12	12	12
	TC	13	13	13	12	12	12	12	12	12
	TCE	12	12	12	12	12	12	12	12	12
MSPLC	TH	12	12	12	12	12	12	12	12	12
	TC	12	12	12	12	13	12	12	12	12
	TCE	12	12	12	12	12	12	12	12	12

The convergence performance of the X-CMFD method is then compared to the MGPI and NOPC-CMFD method. A power relaxation factor of 0.5 is used for the MGPI cases. The WS ratio used for the MGPLC cases is 0.98, while for MSPLC cases 0.96 is used based on the results in Table 5.2. These results are shown in Table 5.3. “OI” is the number of outer iterations to converge the problem and “SC” is the number of shielding calculations. It can be observed that the X-CMFD cases take the smallest number of outer iterations to converge for the P6 problem with different power levels. The number of shielding calculations performed for the X-CMFD cases is close to that for the NOPC-CMFD cases. In terms of run time, overall, X-CMFD cases take a slightly longer time than the NOPC-CMFD cases when the power level is high. The reason is that the NOPC-CMFD method adjusts the WS ratio based on the feedback intensity. For high-power cases, the WS ratio is decreased and the linear system of Eq. (2.47) becomes easier to solve.

$$\mathbf{M} - \lambda_{\mathrm{s}}\mathbf{F}\boldsymbol{\phi} = (\lambda - \lambda_s)\mathbf{F}\boldsymbol{\phi}\,. \qquad \text{(2.47 revisited)}$$

For the X-CMFD method, a fixed WS ratio that is close to 1 is used, therefore, the linear system is more difficult to solve compared to the system of the NOPC-CMFD methods Eq. (2.47). More computation effort is also spent on interpolating the cross sections and updating the CMFD coefficients. Nevertheless, the performance of the X-CMFD method is comparable to that of the NOPC-CMFD method, and the X-CMFD method is much more stable than the current iteration scheme used in MPACT. When power is 140%, the MGPLC cases take at least 24% less run time to converge compared to the MGPI cases. When MSPLC is used, the run time can be reduced by more than 42%.

Table 5.3: Convergence comparison of X-CMFD methods to other CMFD methods for P6

Feedback	CMFD Solver	Power Level (%)									
		20	60	100	120	140	20	60	100	120	140
		OI/SC					**Wall Time (min)**				
TH	MGPI	13/2	13/5	13/6	13/6	15/7	4.4	5.1	5.8	6.0	6.9
	MGPC	13/3	13/5	13/5	13/4	12/4	4.8	5.0	5.1	4.9	4.6
	MSPC	12/2	12/3	13/4	13/5	12/6	3.3	3.5	3.9	4.1	4.1
	MGPLC	12/3	12/4	12/4	12/4	12/4	4.6	4.7	4.8	4.9	5.1
	MSPLC	12/2	12/3	12/4	12/5	12/5	3.4	3.6	3.8	3.9	4.0
TC	MGPI	16/2	13/6	12/7	14/8	15/9	5.8	5.4	5.5	6.5	7.2
	MGPC	12/4	12/5	12/6	13/7	13/7	4.8	4.8	5.0	5.5	5.5
	MSPC	12/3	12/5	12/5	12/6	12/7	3.5	3.9	3.9	4.1	4.3
	MGPLC	12/3	12/5	12/6	12/6	12/7	4.7	5.2	5.5	5.5	5.7
	MSPLC	12/3	12/4	12/6	12/6	12/6	3.6	4.0	4.1	4.2	4.2
TCE	MGPI	12/3	12/5	13/6	15/7	17/10	4.7	5.1	5.9	7.0	8.3
	MGPC	13/3	13/5	13/5	13/4	12/4	4.4	4.8	4.9	5.0	5.1
	MSPC	12/3	12/5	12/5	12/6	12/7	3.6	4.0	4.0	4.2	4.4
	MGPLC	12/3	12/3	12/5	12/5	12/5	4.8	4.9	5.3	5.3	5.4
	MSPLC	12/3	12/4	12/5	12/5	12/5	3.7	4.0	4.1	4.1	4.1

5.4.2.1 Effect of the Shielding Calculation

One drawback of using the NOPC-CMFD is that the selection of the method to estimate the feedback intensity for the partial convergence depends on whether the shielding calculation has been performed for that iteration. The advantage of the X-CMFD is that it does not rely on the estimation of γ, thus the shielding calculation should not affect the performance of X-CMFD.

To confirm this, a series of cases are run and the results are presented in Table 5.4. It can be seen the numbers of outer iterations to converge for X-CMFD cases with different powers are the same as those presented in Table 5.2. Thus, it can be concluded that the presence of the shielding calculation does not affect the performance of the X-CMFD method in P6 cases.

Table 5.4: Number of outer iterations to converge P6 for X-CMFD (shielding calculation enforced)

CMFD Solver	Feedback	Power Level (%)						
		20	40	60	80	100	120	140
MGPLC	TH	12	12	12	12	12	12	12
	TC	12	12	12	12	12	12	12
	TCE	12	12	12	12	12	12	12
MSPLC	TH	12	12	12	12	12	12	12
	TC	12	12	12	12	13	12	12
	TCE	12	12	12	12	12	12	12

5.4.3 Affect of Iterative T/H Solutions

Like with the NOPC-CMFD analysis in Chapter 4, we also wish to investigate the performance of the X-CMFD methods in simulations with more complex TH models that rely on iterative solutions. To do this, MPACT uses CTF [90] for TH feedback. CTF obtains solutions through iterations and is not fully converged. Since the tight convergence criteria for the fission source residual is used, the tolerance for converging the CTF solution is determined by

$$\epsilon_i = \max(1e-6, \epsilon_0 res_f)\,, \tag{5.15}$$

where res_f is the residual of the fission source, and ϵ_0 is the input tolerance for converging the CTF solutions.

The results are shown in Table 5.5. It is again observed that the number of outer iterations in the MSPLC cases is around 12 and not more than 13, the smallest number of outer iterations is 11.

Table 5.5: Number of outer iterations to converge P6 for MSPLC with CTF as the TH solver.

CMFD Solver	Feedback	Power Level (%)						
		20	40	60	80	100	120	140
MSPLC	TH	12	12	12	12	13	12	12
	TC	12	12	12	13	12	11	13
	TCE	12	12	12	12	12	12	12

5.4.4 3×3 Core Problem

The last assessment of the X-CMFD method is performed on the VERA core physics progression problem 4 (P4) a 3×3 core problem. The RCCA is modeled in this problem. Therefore, there is a lot of heterogeneity in axial and radial flux distribution. For P4 cases, the power is 20% and

100% of the rated power, and while we recognize this is not a feasible operational scenario it still presents a challenging problem for the method. The results are shown in Tables 5.6 and 5.7.

Table 5.6: Convergence comparison of the X-CMFD solvers to other CMFD solvers for P4 cases at 100% rated power.

FDB	CMFD Solver	Rod Position							
		23	46	69	92	23	46	69	92
		OI/SC				Wall Time (min)			
–	**Ref**	13	14	13	13	–	–	–	–
TH	MGPI	13/5	13/7	13/7	13/7	13.9	15.1	15.8	14.5
	MGPLC	13/3	13/5	13/4	13/4	14.1	15.7	15.6	15.1
	MGPC	13/5	13/6	14/6	13/6	14.8	15.5	15.7	15.4
	MSPC	13/6	14/6	14/6	13/6	11.2	11.8	11.8	11.2
	MSPLC	13/5	13/6	13/6	13/5	10.9	11.4	11.4	10.9
TC	MGPI	13/6	14/6	13/7	13/6	15.4	16.6	15.9	14.7
	MGPC	13/6	14/6	14/7	14/6	14.5	15.9	15.8	15.1
	MGPLC	13/6	13/6	13/6	13/5	15.3	15.1	15.7	15.0
	MSPC	14/6	14/6	14/7	14/8	11.7	12.3	12.3	12.7
	MSPLC	13/6	13/6	13/6	13/5	11.4	11.4	11.4	10.9
TCE	MGPI	13/7	14/6	13/8	13/6	17.1	17.2	17.5	15.7
	MGPC	13/6	13/6	13/6	13/6	14.4	14.7	14.6	14.7
	MGPLC	13/6	13/7	13/6	13/5	15.7	16.3	15.7	15.3
	MSPC	14/6	13/6	13/6	13/6	12.1	11.5	11.5	11.5
	MSPLC	14/6	13/6	13/6	13/6	12.3	11.7	11.7	11.7

When power is 100% of the total power, it can be seen that the number of outer iterations to converge the MGPI cases is close to that of the "Ref" cases. Therefore, in these P4 cases, the convergence performance of the original iteration scheme is comparable to that of the MGPLC method and the MGPC method. However, since more shielding calculations are performed for the MGPI cases when feedback is from TCE, the run time of the MGPC and MGPLC cases is shorter. For P4 cases with feedback from TCE, the run time for MGPLC cases is reduced by at most 17%. The X-CMFD methods take a smaller number of outer iterations to converge the P4 cases compared to the NOPC-CMFD methods. The run time of the MGPLC cases is longer than that of the MGPC cases when the number of outer iterations to converge is the same. The reason for this is the same as before, in that the linear system of the MGPLC equations is harder to solve. On the contrary, the run time of the MSPC cases and MSPLC cases is very close indicating that the efficiency of solving the linear system with the MSED solver can help to mitigate the performance degradation caused by the aggressive WS ratio. Compared to the MGPI cases, the run time of the MSPLC cases is reduced by at most 33%.

Table 5.7: Convergence comparison of the X-CMFD solvers to other CMFD solvers for P4 cases at 20% rated power.

FDB	CMFD Solver	Rod Position							
		23	46	69	92	23	46	69	92
		OI/SC				Wall Time (min)			
–	**Ref**	13	14	13	13	–	–	–	–
TH	MGPI	16/4	16/6	15/5	14/5	17.5	19.0	16.6	15.3
	MGPC	13/4	14/5	13/5	13/5	13.8	15.1	13.8	13.5
	MGPLC	13/3	14/5	13/4	13/4	14.8	15.5	15.7	15.4
	MSPC	13/4	14/5	13/5	13/6	11.2	11.8	11.8	11.2
	MSPLC	13/3	14/5	13/4	13/4	9.8	11.3	10.3	10.3
TC	MGPI	16/4	16/5	15/5	14/5	17.6	18.4	16.6	15.3
	MGPC	13/5	14/5	13/5	13/5	14.4	15.1	13.8	13.6
	MGPLC	13/3	14/4	13/4	13/4	14.6	16.0	14.6	14.5
	MSPC	13/5	14/5	13/5	13/5	10.7	11.2	10.7	10.7
	MSPLC	13/3	14/4	13/4	13/4	9.9	11	10.3	10.3
TCE	MGPI	13/4	14/4	14/4	13/4	13.6	14.9	14.5	13.7
	MGPC	13/3	13/5	13/5	13/4	12.8	13.9	13.7	13.2
	MGPLC	13/3	13/4	13/3	13/3	14.0	14.3	13.9	13.8
	MSPC	13/3	13/3	13/4	13/3	10.0	10.0	10.5	10.0
	MSPLC	13/3	13/3	13/3	13/3	10.1	10.2	10.2	10.2

When power is 20% of the total power, the MGPI cases take more outer iterations to converge. The X-CMFD cases take the same number of outer iterations as the "Ref" cases. Compared to the run time of the MGPI cases, the run time of MGPLC and MSPLC cases is reduced by at most 17% and 44%, respectively.

5.5 Summary

In this chapter, the X-CMFD method for accelerating solving neutron transport problems with nonlinear feedback has been presented and analyzed via Fourier analysis. The method was implemented in MPACT and optimized based on the current CMFD solvers currently used in MPACT.

First, before introducing the X-CMFD method, the Fourier analysis of a hypothesized iteration scheme, NFCDA, was present. It is an NDA acceleration method, where the low-order diffusion problem is fully coupled with feedback. Fourier analysis results show that the NFCDA in problem with feedback can have comparable performance to the NDA in the problem without feedback.

Then the Fourier analysis of the X-CMFD method was presented. Fourier analysis results suggested that the low-order diffusion eigenvalue problem should be coupled with the feedback, and the feedback is applied in the power iteration of the low-order solver. The new X-CMFD method has significantly improved performance compared to current iteration schemes used in practice. In particular, for scattering-dominated spatially continuous problems, the X-CMFD spectral radius approached the familiar value of $\rho = 0.2247$, while conventional relaxation schemes, such as those studied in [39] diverge for optically thick problems. Another benefit of the X-CMFD method is that it requires no estimation of the feedback intensity. X-CMFD can be understood as a variant of the NFCDA method that solves the fully coupled low-order diffusion problem via Wielandt shifted power iteration.

Next, the prototype X-CMFD that was analyzed from Fourier analysis was implemented in MPACT. Results showed that the X-CMFD method can stabilize the iteration. Therefore, the implementation of X-CMFD was optimized in the way that the feedback is approximated by interpolating the cross sections based on the local fission rate. The optimized X-CMFD was implemented based on the MGPI solver and the MSED solver in MPACT. The two CMFD methods were referred to as the MGPLC method and the MSPLC method.

Finally, numerical results were presented to show the performance of these two implementations for 3D problems with different scales and forms of feedback. The numerical results verified the prediction that the iterative scheme can converge cases faster if the low-order diffusion problem of the X-CMFD solvers is more tightly converged. The convergence performance of the X-CMFD method in problems with feedback is almost the same as that of the fully converged CMFD method in problems without feedback. Compared to the MGPC solver, the MGPLC solver took a longer run time to converge P6 and P4 cases, because the linear system of the low-order problem is harder to solve. Using the MSED method reduced the computational cost of solving the linear system. Therefore, MSPLC cases have comparable run time to MSPC cases.

CHAPTER 6

Transient Multilevel With One-group CMFD Acceleration

The methods presented in the previous three chapters help stabilize the steady-state simulations. Starting from this chapter, the focus is switched on the methods for solving the time-dependent neutron transport equation. In this chapter, a more advanced Transient Multilevel (TML) scheme is developed to speed up the current transient calculation by introducing the one-group CMFD (1GCMFD) system.

In the new scheme, the 1GCMFD equation is utilized to converge the prompt fission source for the 3D-Multigroup CMFD (MGCMFD) system. The prompt fission source term of the MGCMFD system can be treated as a part of the fixed source, and therefore the MGCMFD equation becomes much easier to solve. A dynamic iteration strategy is also developed to minimize the iterations between 1GCMFD and MGCMFD systems. Additionally, the 1GCMFD level is introduced to the previously developed 3-level TML scheme to update the MGCMFD flux via MGCMFD/1GCMFD coupling.

6.1 One Group CMFD Acceleration

The one group or two-group diffusion equation has been well studied and used to accelerate the higher-order MGCMFD calculations in steady-state simulations [102, 103, 32]. These few-group methods can be viewed as an extension of the CMFD method, where a lower-order method is used to accelerate the higher-order method, that is the MGCMFD system. MSED is used to accelerate the steady-state eigenvalue problem [32] by accelerating the convergence of fission source for the steady-state CMFD eigenvalue problems. Following the same idea, the 1GCMFD is used to accelerate the MGCMFD calculations in the transport TFSP.

It is assumed that 1GCMFD is a sufficiently accurate approximation to converge the fission source, and this has been demonstrated for steady calculations [32]. Therefore, the fission source term $\mathbf{F\Phi}$ in the MGCMFD is moved to back the right-hand side (RHS), and the MGCMFD system becomes much easier to solve. Though some computation effort is needed to solve the 1GCMFD system, the number of unknowns in 1GCMFD is much smaller than that in the MGCMFD system. Consequently, the overall computation time to solve the MGCMFD problem is reduced. The numerical results presented later in this chapter will support these claims. It should be noted that, for solving a MGCMFD TFSP system, the 1GCMFD system will be solved for multiple times, and the 1GCMFD coefficients will be updated every time that MGCMFD flux is updated. The details are shown in the following sections.

6.1.1 1G/MGCMFD Iteration

6.1.1.1 Methodology

The general procedure for 1GCMFD acceleration of the MGCMFD TFSP problem is derived in this section.

We already have

$$\begin{gathered}\sum_{m'\in N(m)}\Gamma_{(m,m')}\left[-\tilde{D}^n_{(m,m'),g}(\Phi^n_{m',g}-\Phi^n_{m,g})+\hat{D}^n_{(m,m'),g}(\Phi_{m,g}+\Phi_{m',g})\right]\\ +\left(\Sigma^n_{t,m,g}\Phi^n_{m,g}-\sum_{g'=1}^{G}\Sigma^n_{s0,m,g'\to g}\Phi^n_{m,g'}\right)V_m=\left(\chi^n_{m,g}S^n_{F,m}+S^n_{tr,m,g}\right)V_m\,,\\ 1\le g\le G\,,\quad 1\le m\le M\,.\end{gathered}\tag{2.62 revisited}$$

Eq. (2.62) is integrated over the whole energy phase space and the resulting equation is written as:

$$\begin{gathered}\sum_{m'\in N(m)}\Gamma_{(m,m')}\left[-\tilde{D}^n_{(m,m')}(\Phi^n_{m'}-\Phi^n_{m})+\hat{D}^n_{(m,m')}(\Phi^n_{m}+\Phi^n_{m'})\right]\\ +\left(\Sigma^n_{t,m}\Phi^n_{m}-\Sigma^n_{s,m}\Phi^n_{m}\right)V_m=\left(S^n_{F,m}+S^n_{tr,m}\right)V_m,\quad 1\le m\le M\,.\end{gathered}\tag{6.1}$$

Here Φ^n_m is defined as

$$\Phi^n_m=\sum_{g=1}^{G}\Phi^n_{m,g}\,,\tag{6.2}$$

and the cross section, transient source and spectral terms in Eq. (6.1) are

$$\Sigma^n_{x,m}=\frac{\sum_{g=1}^{G}\Phi^n_{m,g}\Sigma^n_{x,m,g}}{\Phi^n_m}\,,\quad \Sigma_x=A,\ \Sigma_t,\ v\Sigma_f\,,\tag{6.3a}$$

$$S^n_m = \sum_{g=1}^{G} S^n_{m,g} \;\; S = C,\; B\,. \tag{6.3b}$$

This process of obtaining the 1GCMFD coefficients based on the MGCMFD coefficients and solutions is referred to as *collapsing*. And the properties shown in Eqs. (2.38) are not preserved. The procedure to collapse the diffusion coefficients and the nonlinear correction terms is same as that described in [32]. The sum and difference of the diffusion coefficients and the nonlinear correction terms at the surface (m, m') are calculated with the fluxes of cells m and m'as:

$$< \tilde{D}^n_{(m,m')} + \hat{D}^n_{(m,m')} >= \frac{\sum_{g=1}^{G}\left(\tilde{D}^n_{(m,m')} + \hat{D}^n_{(m,m')}\right)\Phi^n_{m,g}}{\Phi^n_m}\,, \tag{6.4}$$

$$< \tilde{D}^n_{(m,m')} - \hat{D}^n_{(m,m')} >= \frac{\sum_{g=1}^{G}\left(\tilde{D}^n_{(m,m')} - \hat{D}^n_{(m,m')}\right)\Phi^n_{m',g}}{\Phi^n_{m'}}\,. \tag{6.5}$$

Then the one group diffusion coefficients and the nonlinear correction terms are obtained from the solution of

$$\tilde{D}^n_{(m,m')} = \frac{< \tilde{D}^n_{(m,m')} + \hat{D}^n_{(m,m')} > + < \tilde{D}^n_{(m,m')} - \hat{D}^n_{(m,m')} >}{2}\,, \tag{6.6a}$$

$$\hat{D}^n_{(m,m')} = \frac{< \tilde{D}^n_{(m,m')} + \hat{D}^n_{(m,m')} > - < \tilde{D}^n_{(m,m')} - \hat{D}^n_{(m,m')} >}{2}\,. \tag{6.6b}$$

It can be shown that with this definition, the properties shown in Eqs. (2.38) are also preserved. Once the 1GCMFD coefficients are calculated, the one group scalar flux is obtained by solving

$$\left(\mathbf{M_1} - \mathbf{S_1} - \mathbf{F_1} - \mathbf{A_1} - \mathbf{B_1}\mathbf{F_1}\right)\mathbf{\Phi_1} = \mathbf{C_1}\,, \tag{6.7}$$

where the subscript $_1$ denotes the 1GCMFD coefficients. Then the MGCMFD flux is calculated by

$$\left(\mathbf{M} - \mathbf{S} - \mathbf{A} - \mathbf{B}\mathbf{F}\right)\mathbf{\Phi} = \mathbf{C} + \chi\mathbf{F_1}\mathbf{\Phi_1}\,. \tag{6.8}$$

The general procedure for 1GCMFD acceleration of MGCMFD is illustrated in Figure 6.1. The 1GCMFD coefficients are collapsed from the MGCMFD coefficients using the latest multigroup flux. The 1GCMFD flux is then obtained by solving Eq. (6.7). And solution is used to update the MGCMFD sources. Eventually, with the fission source from 1GCMFD, the MGCMFD scalar flux is calculated. Since the coefficients in the 1GCMFD equation are collapsed with the MGCMFD flux, that is not a fully converged solution, iterations between the 1GCMFD and MGCMFD systems are advantageous. This iteration process is referred to as 1G/MGCMFD iteration.

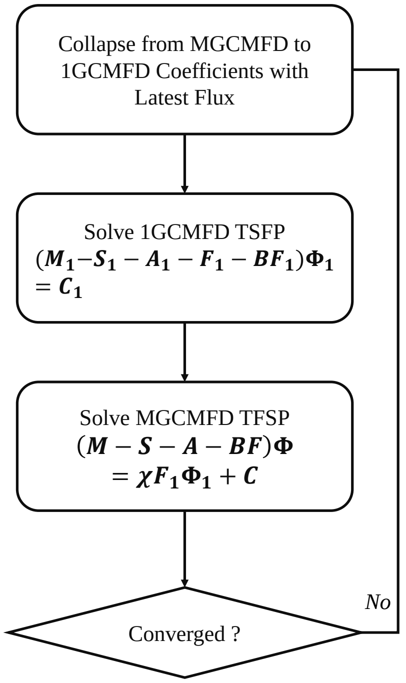

Collapse from MGCMFD to 1GCMFD Coefficients with Latest Flux
Solve 1GCMFD TSFP
$(\boldsymbol{M_1}-\boldsymbol{S_1}-\boldsymbol{A_1}-\boldsymbol{F_1}-\boldsymbol{BF_1})\boldsymbol{\Phi_1} = \boldsymbol{C_1}$
Solve MGCMFD TFSP
$(\boldsymbol{M}-\boldsymbol{S}-\boldsymbol{A}-\boldsymbol{BF})\boldsymbol{\Phi} = \chi\boldsymbol{F_1}\boldsymbol{\Phi_1}+\boldsymbol{C}$
Converged ?
No

Figure 6.1: Flowchart for 1G/MGCMFD iteration.

6.1.1.2 Dynamic 1G/MGCMFD Iteration Strategy

The introduction of the 1G/MGCMFD iteration increases the complexity of the overall acceleration method, and creates an opportunity to perform simulations inefficiently. For example, when the number of 1G/MGCMFD iterations becomes large, the 1G/MGCMFD iteration will becomes more computationally expensive than solving Eq. (2.64) directly. Conversely, if the number of iterations is insufficient at the transport level, the fission source calculated with the 1GCMFD flux solution will not be accurate, and the total calculation time will increase due to the additional transport sweeps required to converge the pin-wise transient source. Moreover, in the MGCMFD level of TML, when the number of the 1G/MGCMFD iterations is insufficient, the multigroup scalar flux of MGCMFD becomes less accurate, and cannot capture the evolution of the amplitude function of the coarse mesh flux.

To remedy these issues, a dynamic iteration strategy to determine the upper limit of the number of the 1G/MGCMFD iterations **nIter** is developed. The idea behind this strategy is to converge the MGCMFD TFSP to a level that is minimally sufficient. When the transport problem becomes easier to solve, the computational effort on the 1G/MGCMFD iteration should be decreased. Meanwhile, if the transport problem becomes harder to solve, the solution of the 1G/MGCMFD iteration should be more tightly converged. The metric for determining the change of the computational

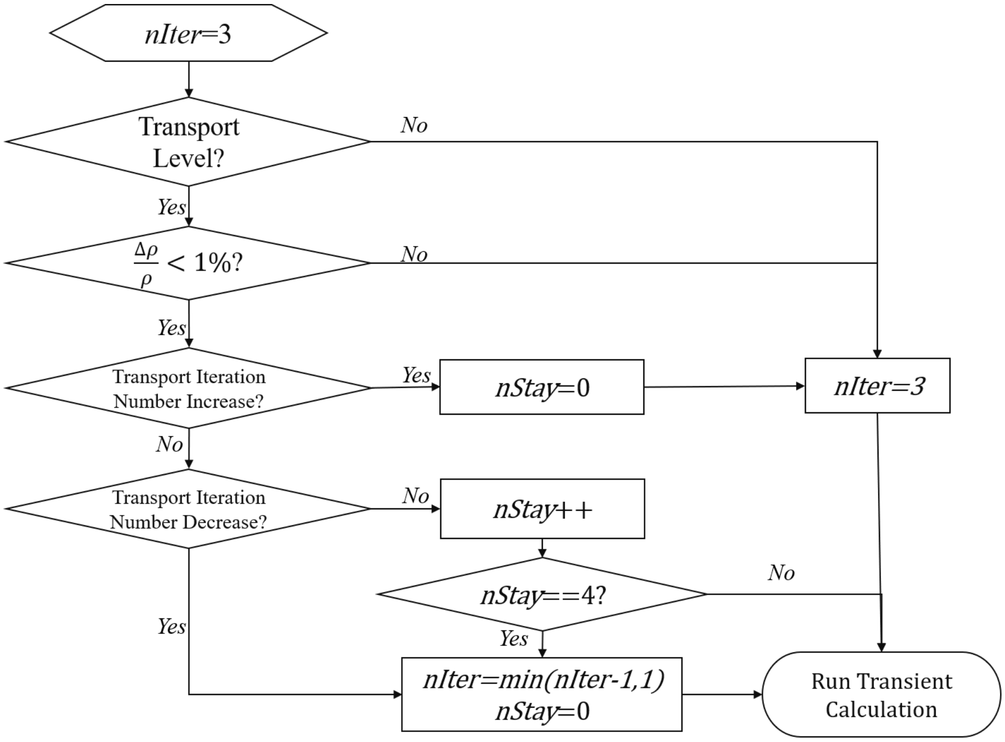


Figure 6.2: Flowchart of dynamic 1G/MGCMFD iteration strategy.

complexity of the transport problem is the variation of the reactivity or the number of transport outer iterations from the previous time steps.

The details of the iteration strategy are illustrated in Figure 6.2. For the 3D-transport level, the initial value of **nIter** is set to be 3. If the number of transport outer iterations stays unchanged for 4 time steps (denoted by **nStay**) or the numbers of transport outer iterations is decreasing for the previous two steps, then **nIter** is decreased by 1. However, **nIter** is set back to 3 when the reactivity changes by 1% or the number of transport outer iteration has increased for the previous two time steps. For the MGCMFD level, **nIter** is set to be 3. The value 3 is used based on the observed numerical result that using three 1G/MGCMFD iterations is a good balance of efficiency and accuracy.

6.1.2 1GCMFD Level in TML

Beyond a simple 1GCMFD acceleration of the MGCMFD TFSP, the TML scheme is modified to include a 1GCMFD level. The new multilevel scheme is referred to as TML-4 because of the 4 computational levels: transport, MGCMFD, 1GCMFD and EPKE. This is illustrated in Figure 6.3.

For the coupling between the MGCMFD level and the 1GCMFD level, two possible approaches may be adopted.

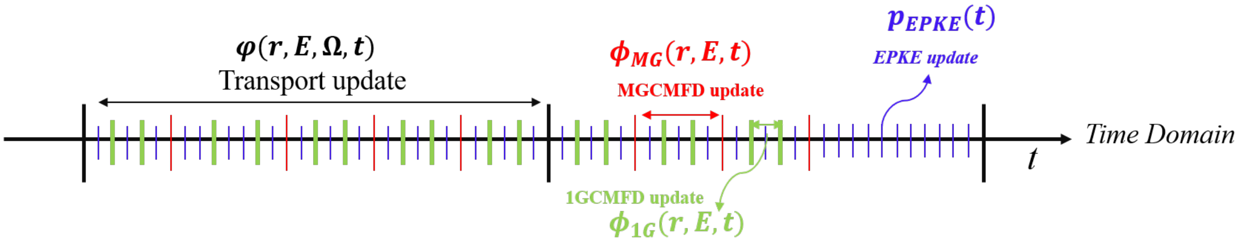


Figure 6.3: TML with one-group CMFD level (TML-4).

- **Flux-multiplication Approach**: The shape function of the coarse mesh scalar flux in energy space is assumed to vary slowly in time, and the amplitude function needs to be corrected by the 1GCMFD calculation. Therefore, the multigroup flux is updated by Eq. (6.9):

$$\phi^{C}_{g,m} = \frac{\Phi^{C}_{m}}{\sum_{g=1}^{G} \phi^{P}_{g,m}} \phi^{P}_{g,m} \,, \tag{6.9}$$

  where $\Phi^{P}_{g,m}$ is the shape of the MGCMFD flux at the predicted MGCMFD step.

- **Source-correction Approach**: The variation of the shape function is not negligible and should be solved with the corrected source that is predicted by the 1GCMFD solutions with Eq. (6.10):

$$\left(\mathbf{M} - \mathbf{S} - \mathbf{A} - \mathbf{BF}\right) \mathbf{\Phi^C} = \mathbf{C} + \chi \mathbf{F_1} \mathbf{\Phi_1^P} \,. \tag{6.10}$$

Numerical results presented later in this chapter will show that the flux-multiplication approach is a better practice to balance the efficiency and accuracy.

The 1GCMFD level and EPKE level are coupled with the aid of the MGCMFD flux. The EPKEs coefficients are calculated with the MGCMFD coefficients, and the solutions that have been corrected using the 1GCMFD-level solution.

The PCQM is as follows:

(i) Solve the MGCMFD TFSP for MGCMFD step $\Delta t^{MG}_{n'}$ using Eq. (2.62). The 1GCMFD coefficients are generated using the equations from Eqs. (6.3) to (6.6).

(ii) Linearly interpolate the 1GCMFD coefficients between $t^{MG}_{n'-1}$ and $t^{MG}_{n'}$ and solve the 1GMCFD problem using the $\Delta t^{1G}_{n''} = \Delta t^{MG}_{n'} / N_{1G}$. N_{1G} is the number of 1GCMFD time steps per MGCMFD time step.

(iii) Update the MGCMFD flux using Eq. (6.9) or Eq. (6.10).

(iv) Update the flux and precursor concentration using the MGCMFD/EPKE coupling with time step $\Delta t^{EPKE}_{n'''} = \Delta t^{1G}_{n''} / N_{EPKE}$.

Here n'' and n''' have been introduced to index the time points on the 1GCMFD level and EPKE level. Figure 6.4 summarizes the TML-4 scheme. The second and third vertical blocks represent the application of 1GCMFD method to accelerate the transient calculation. For the MGCMFD TFSP, the 1GCMFD equation is used to converge the source. In the 1GCMFD level, the 1GCMFD solver is used to reduce the number of MGCMFD time steps. For the TML-4 scheme, the coupling between the neutronics problem and TH problem is not altered, and is the same as the scheme presented in [54].

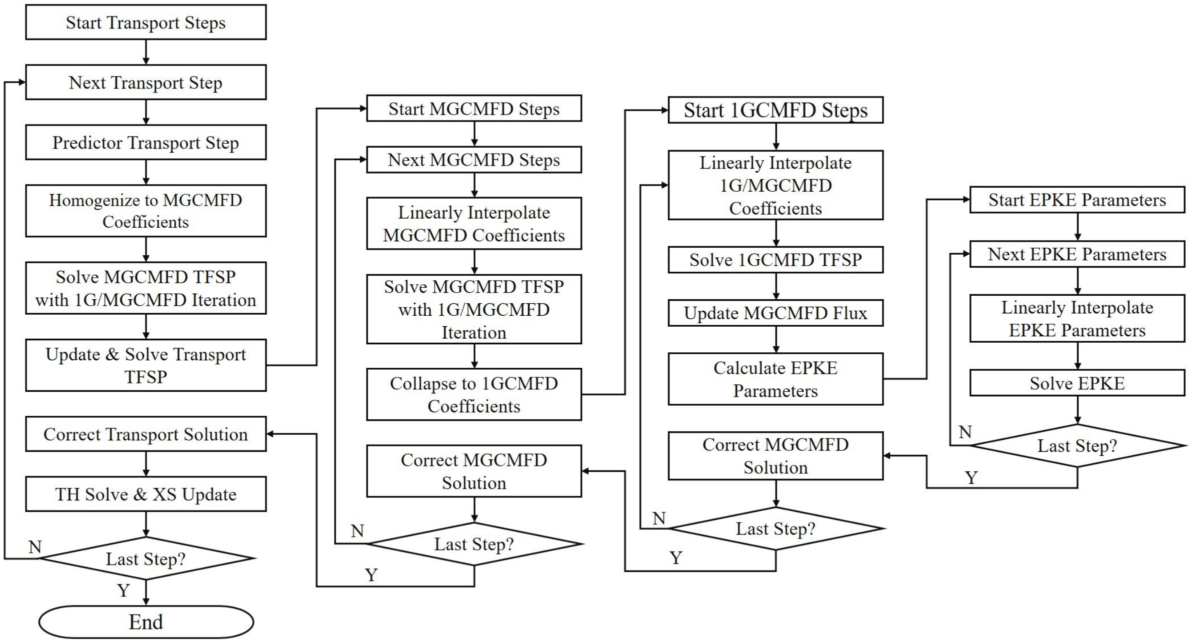

Start Transport Steps
Next Transport Step
Predictor Transport Step
Homogenize to MGCMFD Coefficients
Solve MGCMFD TFSP with 1G/MGCMFD Iteration
Update & Solve Transport TFSP
Correct Transport Solution
TH Solve & XS Update
Last Step?
N
Y
End
Start MGCMFD Steps
Next MGCMFD Steps
Linearly Interpolate MGCMFD Coefficients
Solve MGCMFD TFSP with 1G/MGCMFD Iteration
Collapse to 1GCMFD Coefficients
Correct MGCMFD Solution
Last Step?
N
Y
Start 1GCMFD Steps
Linearly Interpolate 1G/MGCMFD Coefficients
Solve 1GCMFD TFSP
Update MGCMFD Flux
Calculate EPKE Parameters
Correct MGCMFD Solution
Last Step?
N
Y
Start EPKE Parameters
Next EPKE Parameters
Linearly Interpolate EPKE Parameters
Solve EPKE
Last Step?
N
Y

Figure 6.4: Flowchart of TML iteration scheme with 1GCMFD acceleration.

6.2 Numerical Results

The numerical results presented in this section are used to verify the accuracy and the efficiency of utilizing 1GCMFD to accelerate the TML scheme.

Three problems developed based on those presented in [104] – the 4-mini test problem, 7×7 test problem and a full-core problem – are used. The first and second problems are small-sized and medium-sized problems, respectively. The efficiency and accuracy of the 1GCMFD acceleration, and the TML-4 scheme described in this chapter are verified with these two problems. The MGCMFD TFSP Eq. (2.64) of TML-3 can be fully converged within the maximum permitted number of GMRES iterations. Therefore, the MGCMFD TFSP can be considered as accurate, and the simulation results of TML-3 are used as reference results. In MPACT, the default Krylov solver used is the PETSc-based GMRES [96] solver preconditioned by the block Jacobi method. The

GMRES solver is restarted every 100 iterations. The absolute and relative tolerances for solving the linear system are 10^{-10}, and the maximum number of GMRES iterations is 400.

The third problem is a large-scale, full-core problem based on the Watts Bar Unit 1, Cycle 1 [98]. Due to the large size of Eq. (2.64) and the degradation of the Krylov solver, the linear system cannot be fully converged with the user-specified number of GMRES iterations. As a result, the problem is used to show the efficiency of the 1GCMFD acceleration and the TML-4 scheme in a practical reactivity insertion accident (RIA) simulation of the full-core reactor.

For all three problems, the results from the transient calculation without TML but with finer steps will not be used as the reference results to verify the accuracy of the method, because the accuracy of the TML scheme has been investigated in [54] and [50]. Additionally, the focus of the research is to develop the methods to improve the efficiency of the TML scheme.

6.2.1 4-mini Test Problem

6.2.1.1 Problem Details

The 4-mini test problem is a 3D regression test problem in MPACT used to verify the implementation of the transient methods. The problem has the same layout with VERA Progression Problem 4 (P4) [98] but with 7×7 assemblies. The core geometry is shown in Figure 6.5a. The assemblies in red are the regions with 2.11% enriched UO_2 and the assemblies in green are the regions with 2.6% enriched UO_2 and with 4 Pryex rods inserted. The corner red assemblies contain 4 stainless steel rods inserted in the empty locations. The hybrid AIC/B_4C RCCA is located in the center assembly. Unless specified, the composition of the materials and the state variables (inlet temperature, boron concentration, etc) used in the 4-mini test problem are the same as those used in P4. Compared to P4, the height of the 4-mini test problem is reduced. The fuel stack is adjusted to be 209.16 cm rather than 365.76 cm in P4. The poison height of the AIC and B_4C is adjusted to be 58.024 cm and 147.96 cm. The rated power is 19.36 MW, and the rated flow is 131.16 kg/s.

For the initial condition, the RCCA is fully inserted. The transient cases with initial hot zero power (HZP) and hot full power (HFP) conditions are simulated for 1.0 s, with RCCA withdrawn out of core over a distance of 36.27 cm in 0.05 s. The initial power of the HZP case is 0.0136% of the rated power, and the initial power of the HFP case 100% of the rated power.

Seven different cases are run to verify the 1G/MGCMFD iteration and the new TML-4 scheme. All the cases are run with a 5 ms step size for the transport level. Cases 1 through 4 are used to assess the performance and accuracy when only the 1G/MGCMFD iteration with the dynamic iteration strategy is used to accelerate the original TML-3 scheme. 10 finer steps in the MGCMFD level are used for the solution of the reference case (Case 1). Case 2 uses 1G/MGCMFD iteration to solve the MGCMFD TFSP with the same time discretization as Case 1. Case 3 uses 5 MGCMFD

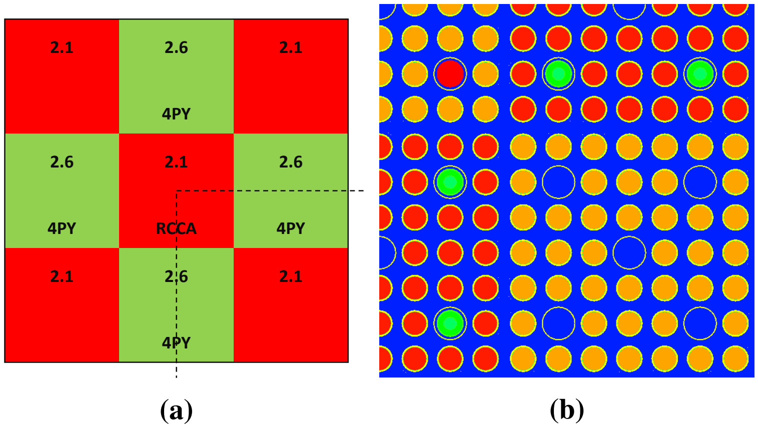


Figure 6.5: Assembly layout and geometry of the 4-mini test problem. (a) shows the assembly, control rod bank and poison of the problem. (b) shows the geometry of the 4-mini test problem.

time steps per transport step that is the default TML-3 option in MPACT. Case 4 has the same time discretization as Case 3, but uses the new 1G/MGCMFD iteration acceleration.

The performance and accuracy of the new TML-4 scheme are investigated by comparing the results of Case 5 through 7 with Case 3. Case 5 and Case 6 use 3 finer steps in both the MGCMFD and the 1GCMFD levels, while Case 7 only use 3 finer steps in the MGCMFD level. The flux-multiplication approach is used to correct the multigroup flux in Case 5, while in Case 6 the source-correction approach is used. The details of these cases are summarized in Table 6.1, where “nMG” stands for the number of MGCMFD steps per transport time step, “n1G” stands for the number of 1GCMFD steps per MGCMFD step, and “nEPKE” denotes the number ofEPKE steps per CMFD step.

All the cases are performed with the Chebyshev-Yamamoto quadrature set with 16 azimuthal angles and 2 polar angles per octant. The spatial discretization use 8 azimuthal flat source regions per pin-cell, 3 radial rings in the fuel, and 1 ring in the gap, cladding and moderator separately. The P_3 axial solver is used for this case with 30 axial layers. The cases are run on the Lighthouse cluster with 16 cores (Xeon E5-2680v3 processor) at the University of Michigan. The cross sections for MPACT are provided from a 51-group library generated at ORNL by CASL [68].

6.2.1.2 Power and Reactivity Results

The run time results are shown in Table 6.2. The run time is given in hours. The following abbreviations are used to condense the column headers. “TR” gives the total run time required by the simulation, “CR” presents the run time for the CMFD solver, and “NCR” shows the run time of the non-CMFD solvers. “TR RD” and “CR RD” represents the run time reduction compared to the time of the reference cases for “TR” and “CR” respectively.

The run time differences between Case 1 (TML-3-ref) and Case 2 (TML-3-ref/1G) or Case 3 (TML-3) and Case 4 (TML-1G) show that the CMFD time can be reduced by more than 48% and the total run time can be reduced by around 17% by using the 1G/MGCMFD iteration. Moreover, the non-CMFD time varies little when the 1G/MGCMFD iteration is applied, showing that the dynamic iteration strategy does not degrade the convergence rate of the fine mesh transport flux solution. The acceleration performance from 1G/MGCMFD is slightly less when the initial condition is HFP compared to HZP. One reason is that the CMFD solver run time represents a smaller portion in the total run time for the HFP cases as more time is used in solving the TH equations, updating the cross sections, etc. The other reason is that the CMFD solver run time reduction in the HFP cases is also slightly less.

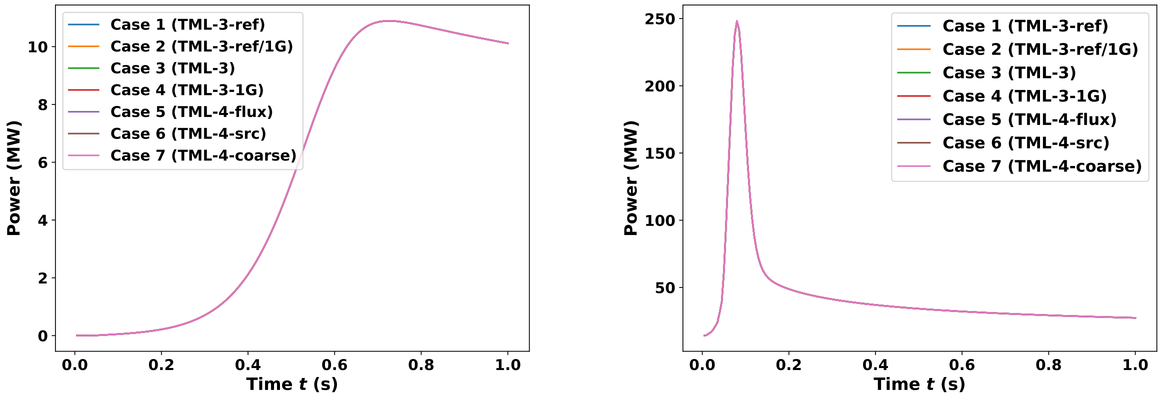
Table 6.1: TML options for seven different cases based on the 4-mini test problem.

Case	Name	nMG	n1G	nEPKE	1G/MGCMFD Iteration	Remark
1	TML-3-ref	10	—	10	No	Solution reference
2	TML-3-ref/1G	10	—	10	Yes	1G/MGCMFD iteration used
3	TML-3	5	—	10	No	Default TML-3 option in MPACT
4	TML-3-1G	5	—	10	Yes	Default TML-3 option with 1G/MGCMFD iteration
5	TML-4-flux	3	3	10	Yes	TML-4 scheme (flux-multiplication)
6	TML-4-src	3	3	10	Yes	TML-4 scheme (source-correction)
7	TML-3-coarse	3	—	10	No	TML-3 scheme with fewest MGCMFD steps
8	TML-4-hybrid	5	—	10	Yes	Control rod moving
		3	3	10	Yes	Stationary control rod

Table 6.2: Run time for different cases based on the 4-mini test problem.

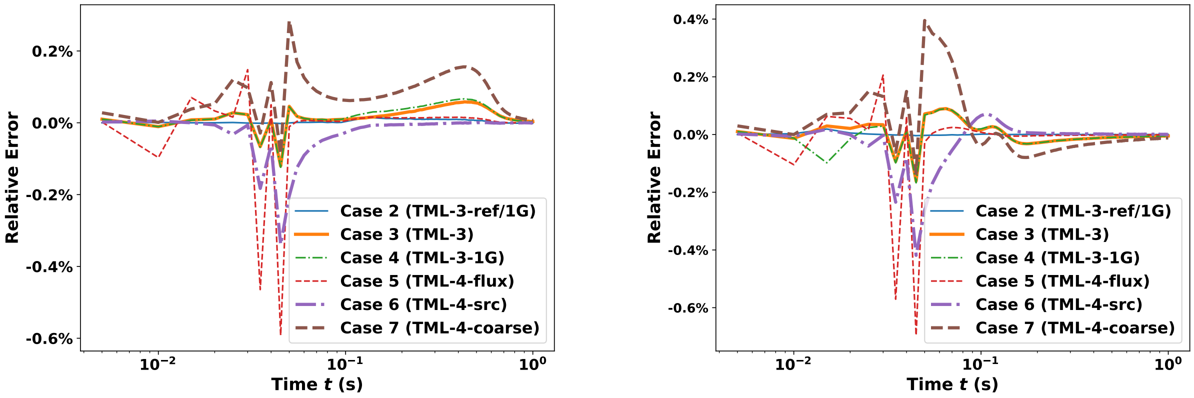
Case	HZP					HFP				
	TR (hrs)	CR (hrs)	NCR (hrs)	TR RD	CR RD	TR (hrs)	CR (hrs)	NCR (hrs)	TR RD	CR RD
1 (Time Ref)	5.26	2.25	3.02	N/A	N/A	5.76	2.39	3.37	N/A	N/A
2	4.18	1.149	3.03	21%	49%	4.58	1.25	3.33	20%	48%
3 (Time Ref)	4.60	1.60	3.00	N/A	N/A	5.04	1.7	3.34	N/A	N/A
4	3.73	0.70	3.04	19%	56%	4.16	0.82	3.34	17%	52%
5	3.86	0.67	3.19	16%	58%	4.22	0.75	3.46	16%	56%
6	4.14	0.98	3.16	10%	39%	4.52	1.05	3.47	10%	38%
7	4.29	1.31	2.98	7%	18%	4.75	1.37	3.38	6%	19%
TML-4-hybrid	3.75	0.71	3.04	18%	56%	4.15	0.81	3.34	18%	52%

The run time differences between Case 3 (TML-3) and Case 5 (TML-4-flux) show that the new TML-4 scheme can reduce the CMFD run time by more than half when using Eq. (6.9) to correct



(a) **(b)**

Figure 6.6: Power history for different cases of the 4-min problem. (a) shows the power for the HZP cases. (b) illustrates the power for HFP cases. In each figure, the power for each case is overlapping each other.



(a) **(b)**

Figure 6.7: Error of the power results of different 4-mini test problem cases. (a) shows the results for the HZP cases while, (b) illustrates the power for the HFP cases.

the multigroup flux. Because the TML-4 scheme solves fewer MGCMFD TFSP linear systems, but solves more 1GCMFD TFSPs, the run time of the TML-4 scheme is close to the run time of using pure 1G/MGCMFD iteration to accelerate the default TML-3 scheme. However, the run time of the TML-4 case is slightly longer than that of the default TML-3 scheme with the 1G/MGCMFD acceleration due to additional computational effort spent on calculating the reactivity, interpolating coefficients, building linear system, etc. The run time of the TML-4 scheme with Eq. (6.10) is longer than the TML-3 scheme with 10 finer MGCMFD steps accelerated by the 1G/MGCMFD iteration, indicating much more time has been spent on solving the additional MGCMFD system of Eq. (6.10).

In terms of the accuracy, Figures 6.6 show that the power history for different cases with the same initial condition are almost identical. This indicates that the TML scheme is a very efficient method to capture the evolution of the amplitude and the shape functions. The relative error of the power solutions in Cases 2 through 7 compared to the reference solution is calculated in order to more thoroughly quantify the accuracy of the different methods. These comparisons are plotted in Figure 6.7. The logarithmic scale is used for the x-axis to better resolve the relative error when the RCCA is being moved. The L_∞ norm and L_1 norm, i.e., the maximum and the average of the relative error of power solutions in magnitudes for both the control rod withdrawal stage and the feedback stage are summarized in Table 6.3.

The L_∞ norm of the relative error for the power solution of Case 2 (TML-3-ref/1G) is the smallest. This norm is smaller than 2×10^{-4}, showing that the error from the 1G/MGCMFD acceleration with the dynamic iteration strategy is negligible, and does not affect the accuracy of the final solution. The same conclusion is also drawn by examining the norms of the relative error for Case 3 (TML-3) and Case 4 (TML-3-1G). These are very close in Table 6.3 and plots of relative errors in Figures 6.7 are overlapping each other.

For the TML-4 scheme, the relative errors of power solutions using the flux-multiplication approach are larger than those for TML-3 with 5 MGCMFD steps or even with 3 MGCMFD steps when the control rods are being withdrawn out of the core. The source-correction approach reduces the error, but both norms of the relative error are still larger than those of the results in Case 7 (TML-3-coarse). As mentioned above, Case 6 (TML-4-src) takes longer to run than Case 2 (TML-3-ref/1G). Therefore, the source-correction approach is not a good choice. When the control rods stop moving, feedback dominates the reactivity change. The amplitude of the relative error of TML-4 decreases rapidly and eventually becomes smaller than the relative error of the power in Case 3 (TML-3) and Case 4 (TML-3-1G). These results indicate the assumption that the shape function in the energy space varies more slowly than the energy-integrated function is not sophisticated enough for transient processes involving control rod movement, but it is very accurate when reactivity changes more slowly.

In light of these results, a more advantageous way to use the 1GCMFD solution to accelerate the TML scheme is to apply the 1G/MGCMFD acceleration with the default TML-3 settings when the control rods are moving, and then to use TML-4 with 3 finer MGCMFD steps per transport step and 3 finer 1GCMFD steps per MGCMFD step with the flux-multiplication method to correct the multigroup flux. This method is referred to as the hybrid TML-4 scheme. The details and the run time are summarized in the last row of Table 6.1 and Table 6.2, respectively. The relative error of the solutions of hybrid TML-4 scheme is compared with the relative error of solutions of the

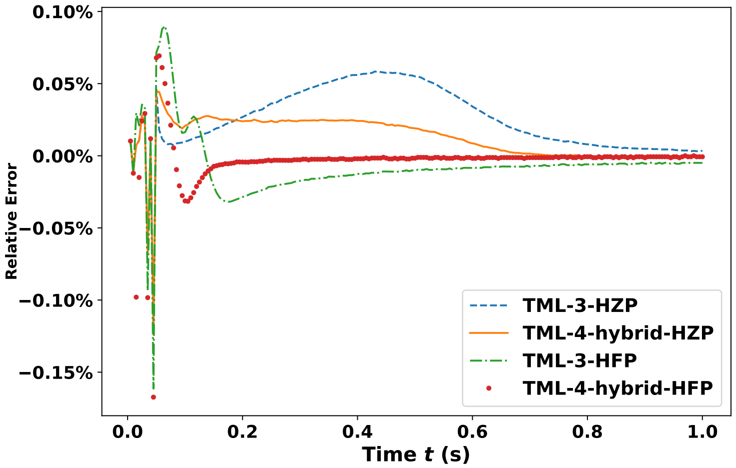


Figure 6.8: Comparison for the core power of the 4-mini test problem with the hybrid method and the default TML-3 method.

default TML-3 settings and the result is plotted in Figure 6.8. The L_1 and L_∞ norms of the error are also summarized in Table 6.3. It can be seen that the relative error of results for the control rod movement part has been reduced, and the L_1 and L_∞ norms of the relative error is approximately the same order as those in the TML-4 cases after the control rod movement. Additionally, the run time and CMFD time are very close to those of TML-4 with the flux-multiplication approach or TML-3 with 1G/MGCMFD acceleration.

To summarize, the results from the simple 4-mini regression test problem show that utilizing the 1G/MGCMFD iteration to accelerate the MGCMFD TFSP calculation can reduce the CMFD time by more than 48% and the total run time by more than 17%. Using the flux-multiplication approach in the new TML-4 scheme can achieve a similar speedup, and have much more accurate power solutions when feedback dominates the evolution of the reactivity. TML-4 with the source-correction approach does better to capture the evolution of power than the flux-multiplication approach when the control rods are being withdrawn from the core, but can not achieve the same accuracy as TML-3 with 1G/MGCMFD acceleration. The source-correction is also slower than the TML-3 scheme with 10 finer MGCMFD steps accelerated by 1G/MGCMFD iterations. A more advantageous iteration scheme is proposed by using TML-3 with 1G/MGCMFD to simulate the transient process when the control rods are moving, and using TML-4 with the flux-multiplication method to simulate the transient process when feedback dominates the change of the reactivity. For the following cases, the performance of the TML-4 scheme with source-correction approach will not be investigated, due to its drawback in balancing the accuracy and efficiency observed in this problem.

Table 6.3: L_1 and L_∞ norm of the relative error for the core power solutions the of 4-mini test problem.

Case	HZP				HFP			
	Rod Movement		Feedback Stage		Rod Movement		Feedback Stage	
	L_1	L_∞	L_1	L_∞	L_1	L_∞	L_1	L_∞
2 (TML-3-ref/1G)	7.00E-06	2.13E-05	5.03E-05	1.41E-04	3.90E-05	1.91E-04	2.60E-06	2.27E-05
3 (TML-3)	3.32E-04	1.21E-03	2.71E-04	5.84E-04	4.83E-04	1.61E-03	1.40E-04	9.00E-04
4 (TML-3-1G)	3.33E-04	1.24E-03	3.14E-04	6.67E-04	5.34E-04	1.67E-03	1.43E-04	9.00E-04
5 (TML-4-flux)	1.48E-03	5.92E-03	7.62E-05	1.60E-04	1.81E-03	6.96E-03	2.63E-05	2.46E-04
6 (TML-4-src)	8.41E-04	3.39E-03	4.79E-05	1.25E-03	1.07E-03	4.19E-03	7.33E-05	1.69E-03
7 (TML-3-coarse)	8.50E-04	2.85E-03	7.82E-04	1.70E-03	1.20E-03	3.95E-03	3.54E-04	3.50E-03
8 (TML-4-hybrid)	3.33E-04	1.24E-03	1.36E-04	4.43E-04	5.34E-04	1.67E-03	4.39E-05	6.93E-04

6.2.2 7×7 Test Problem

The 7×7 test problem is used to assess the performance of the 1G/MGCMFD iteration and TML-4 in a more complex problem with multiple assemblies. The test problem is a hypothetical small modular reactor (SMR) 7×7 core with 37 standard Watts Bar Unit 1, Cycle 1, 17×17 assemblies. The layout is illustrated in Figure 6.9, which consists of three regions of assemblies with specific enrichments. The green region represents the 2.11% enriched UO_2, the red region is the 2.62% enriched region and the blue region is 3.10% enriched. Three different RCCA control banks are loaded, with index A, B and T. More details about the modeling of the fuel pins, Pyrex rods, assemblies, and control banks can be found in Ref [98].

Table 6.4: L_1 and L_∞ norms of the relative error for the core power and reactivity solution of the 7×7 test problem.

Case	Power				Reactivity			
	Rod Movement		Feedback Stage		Rod Movement		Feedback Stage	
	L_1	L_∞	L_1	L_∞	L_1	L_∞	L_1	L_∞
2 (TML-3)	1.67E-03	3.33E-03	3.92E-03	1.52E-02	1.31E-03	4.56E-03	4.81E-04	2.01E-03
3 (TML-3-1G)	1.58E-03	3.05E-03	3.59E-03	1.38E-02	1.11E-03	4.14E-03	4.31E-04	1.83E-03
4 (TML-4-flux)	5.32E-03	1.23E-02	3.19E-03	1.15E-02	2.18E-03	7.01E-03	1.87E-04	1.08E-03
5 (TML-4-hybrid)	1.58E-03	3.06E-03	9.13E-04	2.94E-03	1.11E-03	4.14E-03	5.10E-05	3.20E-04

For the initial condition, only the central control bank, T, is fully inserted into the core region. The other two control banks A and B are fully withdrawn. The transient simulation models the bank T being fully ejected from the core in 0.1 s. The maximum reactivity inserted is $1.2. The transient process is simulated for 0.8 s. During the whole simulation, Bank A and B remain fully out of the core.

Five different cases are run. Case 1 is the reference solution using TML-3 with the same parameters as those in Case 1 of the 4-mini test problem. Case 2 uses the default TML-3 scheme, and Case 3 uses the same time discretization but adds the 1G/MGCMFD iteration to accelerate the MGCMFD calculation. Case 4 and Case 5 adopt the new TML-4 scheme with 3 finer MGCMFD steps, 3 finer 1GCMFD steps and 10 finerEPKE steps, and the flux-multiplication approach to correct the multigroup flux. The difference between Case 4 and Case 5 is that Case 5 uses the new hybrid TML-4 scheme. All the cases are run with 32 processors on the Lighthouse cluster at the University of Michigan, with a transport time step size of 5 ms. The details of the cases are summarized in Table 6.5.

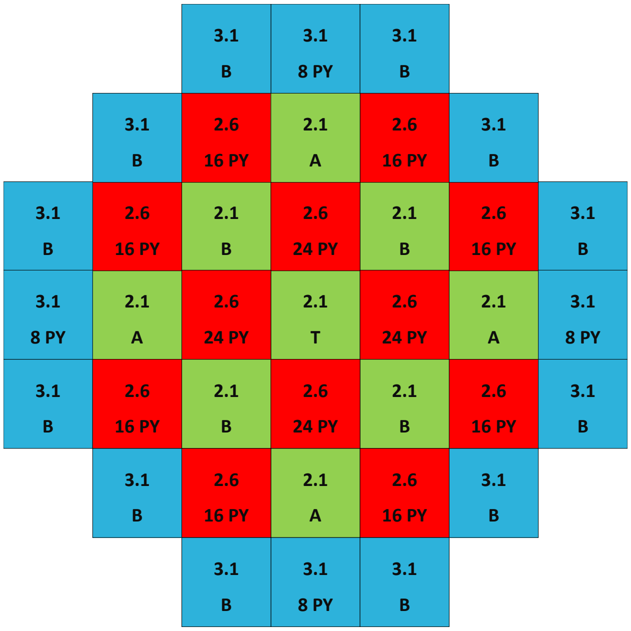


Figure 6.9: Geometry of the 7×7 test problem.

The power history, reactivity history and the run time are shown in Figure 6.10a, Figure 6.10b and Table 6.6, respectively. The results are consistent with the findings in the 4-mini test problem. The power history and reactivity history of different cases are similar to each other. The 1G/MGCMFD acceleration, the new TML-4 scheme, and the hybrid TML-4 scheme also show similar performance. The 1G/MGCMFD acceleration and the TML-4 scheme can both reduce the CMFD time by at least 58% and the total run time by more than 18%.

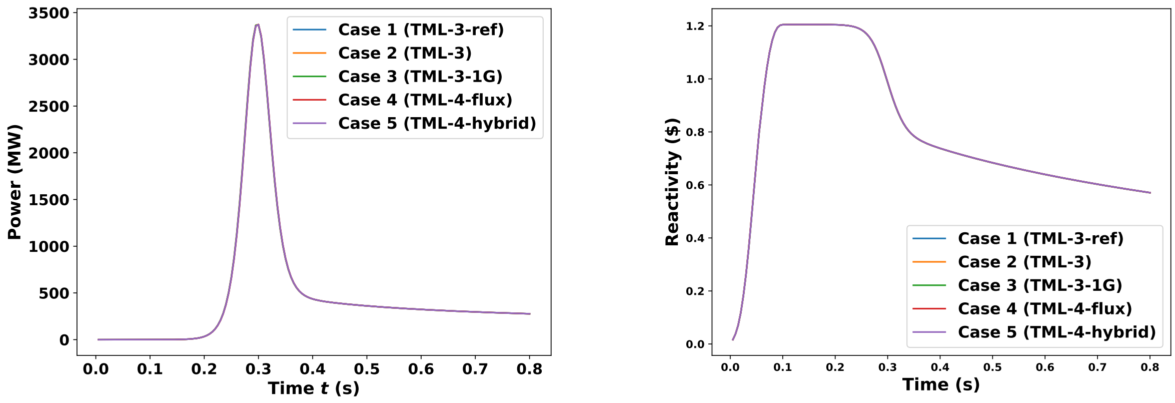
In terms of accuracy, as shown by Figure 6.11a, Figure 6.11b and Table 6.4, the relative errors of the power and reactivity using the TML-4 scheme with the flux-multiplication are the largest with the control rods moving. The L_∞ norm of the relative error is larger than 1%. During the

Table 6.5: Parameters for five different cases based on the 7×7 test problem.

Case	Name	nMGCMFD	n1GCMFD	nEPKE	1G/MGCMFD Iteration	Remark
1	TML-3-ref	10	—	10	No	
2	TML-3	5	—	10	No	
3	TML-3-1G	5	—	10	Yes	
4	TML-4-flux	3	3 (flux-multiplication)	10	Yes	
5	TML-4-hybrid	5	—	10	Yes	Rod Movement
		3	3 (flux-multiplication)	10	Yes	Stationary Rod

Table 6.6: Run time for different cases based on the 7×7 test problem.

Case	TR (hrs)	CR (hrs)	NCR (hrs)	TR RD	CR RD
1	52.62	24.70	27.91	N/A	N/A
2 (Time Ref)	46.07	17.57	28.50	N/A	N/A
3	35.85	7.22	28.63	20%	58%
4	36.27	6.43	29.84	18%	63%
5	36.33	6.65	29.79	18%	62%

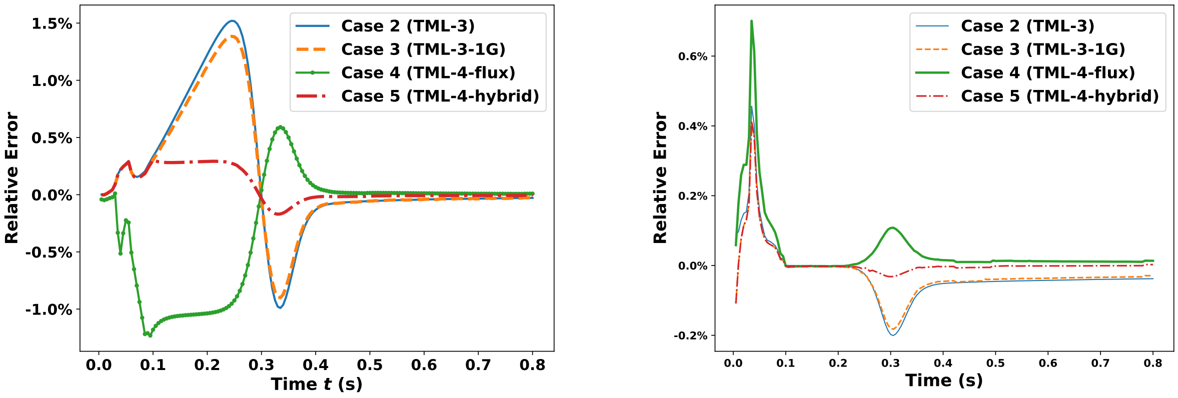


(a)

(b)

Figure 6.10: Power (a) and reactivity (b) histories for the 7×7 test problem. The plots of reactivity and power for all cases are overlapping each other respectively.

time range 0.1 s–0.2 s in the thermal feedback stage, the relative errors of the TML-4 power results change little while the relative errors in the power results of the TML-3 cases keep increasing and become larger than the relative error of the TML-4 results. This observation indicates that the differences compared to the reference solutions are mainly caused by the differences accumulated



(a)

(b)

Figure 6.11: Relative errors of power history and reactivity history of the 7×7 test problem. (a) shows the relative error of power. (b) shows the relative errors of the reactivity history.

at the control rod movement stage for the TML-4 schemes, and by the time discretization in the default TML-3 scheme. The amplitude of the relative error of the default TML-3 scheme with 1G/MGCMFD acceleration is smaller than that for the default TML-3 scheme without acceleration. The norms of the relative error of both solutions are in close agreement, supporting the conclusion that the error introduced by 1G/MGCMFD iteration is negligible.

The L_∞ and L_1 norms of the power results are larger for the default TML-3 scheme compared to the TML-4 scheme when the control rods stop moving. Therefore, it can again be concluded that the TML-4 scheme does better to capture the power evolution when the cross sections vary slowly. This conclusion is also supported by the results in illustrated in Figure 6.11b and Table 6.4 that relative errors of the reactivities for Case 4 (TML-4-flux) and Case 5 (TML-4-hybrid) are smaller than that for Case 2 and Case 3, when the control rods stop moving.

The proposed hybrid TML-4 scheme, has the smallest relative error, and almost the same performance as 1G/MGCMFD iteration to accelerate the TML-3 scheme with default settings, or the TML-4 scheme.

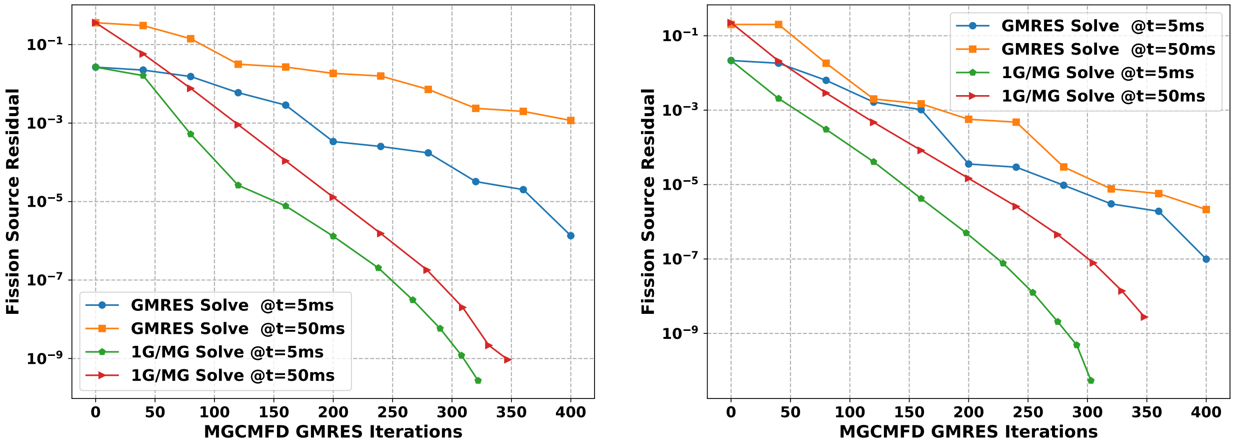
6.2.3 Efficiency of 1GCMFD Acceleration

The numerical results in the previous sections have shown that 1GCMFD acceleration can reduce the MGCMFD calculation time significantly. In this part, more details are provided to support the efficiency of the 1GCMFD acceleration.

Table 6.7 summarizes the average number of GMRES iterations to solve the linear system Eq. (2.64) in the TML-3 case and the linear system of Eq. (6.8) in the TML-3-1G case. As mentioned above, the Krylov solver used is the Block Jacobi preconditioned GMRES solver. It can be seen that when the fission source is moved to the RHS of the system, the number of GMRES iterations to solve the MGCMFD system for the 4-mini test problem and 7×7 test problem can be reduced by more than 80%.

Table 6.7: Average number of GMRES iterations to solve the MGCMFD TFSP system.

4-Mini HZP		4-Mini HFP		7×7 Test Problem	
TML-3	TML-3-1G	TML-3	TML-3-1G	TML-3	TML-3-1G
353	56	356	59	330	49



(a)

(b)

Figure 6.12: Convergence of the fission source for the Krylov solution method and the 1G/MGCMFD iteration method. (a) shows the results for the 4-mini test problem, while (b) illustrates the results for the 7×7 test problem. The plot shows that the convergence rate of 1G/MGCMFD iteration approach is much faster than that of solving the MGCMFD system using Krylov Method.

Table 6.8: Summary of run time and iteration number for solving an MGCMFD TFSP system

State Point (ms)	4-mini HFP				7×7 Test Problem			
	Krylov		1G/MG		Krylov		1G/MG	
	Iter #	Run Time (s)	Iter # MG/1G	Run Time (s)	Iter #	Run Time (s)	Iter # MG/1G	Run Time (s)
5	400	3.6	322/979	2.3	400	51	303/971	38
50	400	3.6	347/997	2.5	400	51	322/1000	38

The efficiency of the 1GCMFD acceleration is further illustrated by comparing the convergence of the fission source norm of the two iterative methods, namely solving the MGCMFD TFSP directly with a Krylov method (Krylov solve) and the 1G/MGCMFD iteration (1G/MG solve). For consistency, the relative error of the fission source norm is calculated. The reference fission source distribution is obtained by solving Eq. (2.64) using a Krylov solver directly with the tolerance set to be 10^{-10}.

For the Krylov method, Eq. (2.64) solution is obtained every 40 GMRES iterations. The fission source relative error is then calculated. For consistency, the maximum number of iterations for solving the MGCMFD system in the 1G/MG solve is also set to 40, and the number of 1G/MGCMFD iterations is 10. The fission source relative error is obtained, and the actual number of GMRES iterations is recorded every 1G/MGCMFD iteration. The MGCMFD TFSP problem solved is the first MGCMFD TFSP problem at each time step. The results are shown in Figures 6.12. It is observed that for both the 4-mini HFP test problem and the 7×7 test problem, the 1G/MGCMFD iteration converges the fission source much faster than the Krylov solver. The 1G/MGCMFD can take around half of the number of GMRES iterations but achieve a similar accuracy compared to that of the Krylov solve method.

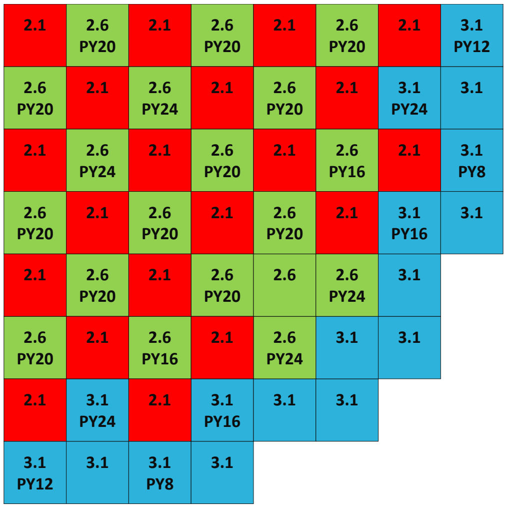
Table 6.8 summarizes the run time and the total number of GMRES iterations for solving the MGCMFD TFSP investigated in Figures 6.12. It can be observed that the 1G/MGCMFD iteration takes both fewer iterations and less time. The time reduction observed here is more than 25%. However, in the numerical simulations performed in the previous two subsections, the MGCMFD TFSP is solved with up to 3 1G/MGCMFD iterations. Therefore, better efficiency from the 1GCMFD acceleration has been observed in Tables 6.2 and 6.6.

6.2.4 Full-core Hot Full Power RIA Problem

In this section, we examine a realistic full-core PWR reactivity insertion accident problem. The model proposed is based on the end-of-cycle (EOC) condition of Problem 9 (P9) in the VERA Progression Problem suite [98]. The fuel assembly and poison layouts are shown in Figure 6.13 in quarter symmetry. The control rod layout is shown in Figure 6.14 in full symmetry. More details such as materials, geometry, and operating condition can be found in [98]. The simulation of the problem is performed with the full geometry, not quarter-core symmetry.

For the transient simulation, the initial condition is at full power, with control rod bank D fully inserted. The control rod bank C is withdrawn to 112 steps based on rod insertion limits. All the other control rod banks are fully withdrawn from the core. The RCCA located at D12 represents the highest worth rod in the core and is assumed to be ejected out of the core in 0.1 s at a constant speed. During the rod ejection, the SCRAM signal is assumed to be triggered to trip the core.

However, the RCCA labeled with "STK" in control bank SD is assumed stuck out of the core during the SCRAM. The transient process is very complex because of the control rod ejection, reactor trip, SCRAM, and stuck control rod.



2.1	2.6 PY20	2.1	2.6 PY20	2.1	2.6 PY20	2.1	3.1 PY12
2.6 PY20	2.1	2.6 PY24	2.1	2.6 PY20	2.1	3.1 PY24	3.1
2.1	2.6 PY24	2.1	2.6 PY20	2.1	2.6 PY16	2.1	3.1 PY8
2.6 PY20	2.1	2.6 PY20	2.1	2.6 PY20	2.1	3.1 PY16	3.1
2.1	2.6 PY20	2.1	2.6 PY20	2.6	2.6 PY24	3.1	
2.6 PY20	2.1	2.6 PY16	2.1	2.6 PY24	3.1	3.1	
2.1	3.1 PY24	2.1	3.1 PY16	3.1	3.1		
3.1 PY12	3.1	3.1 PY8	3.1				

Figure 6.13: Assembly and poison layout of the full-core problem (quarter symmetry). This is regenerated from [98]

.

During the rod ejection, the trip of the reactor would be triggered when the total core power reaches 112% of the rated full power (fp), or the rate of change in the total power is larger than 7.5% fp/s. All the control rods are then inserted assuming a delay. This delay is 0.4 s later from when the SCRAM is triggered by the total power and 0.3 s later if the SCRAM is triggered by the power changing rate. The control rods are fully inserted in 5.5 s. For the first 2.2 s, the insertion rate is 70 steps/s. For the later 3.3 s, the insertion rate is 20 steps/s. The length of the step size of rod movement can be found in [98].

The transient process is simulated for 5 s and divided into 7 periods. The time step sizes used in each period are summarized in Table 6.9. It is noted that the time step sizes are adopted by the CASL Team to illustrate the capability to perform large-scale transient simulations. For practical nuclear reactor simulation, a sophisticated investigation on the time step size is required. However, this is a focus in later chapters albeit for smaller problems.

The problem is simulated on 6264 cores on the Falcon high performance computing system of the Idaho National Laboratory with nodes used exclusively by the job. Three different cases are performed. The first case TML-3 uses the default TML option for MPACT. The 1G/MGCMFD acceleration is used in the second case TML-3-1G. The third case TML-4 uses the TML-4 configuration. For both TML-3-1G and TML-4, the multigrid solver [32] is also used to solve the CMFD linear system, to see the synergy of both the 1GCMFD acceleration and the multigrid solver in

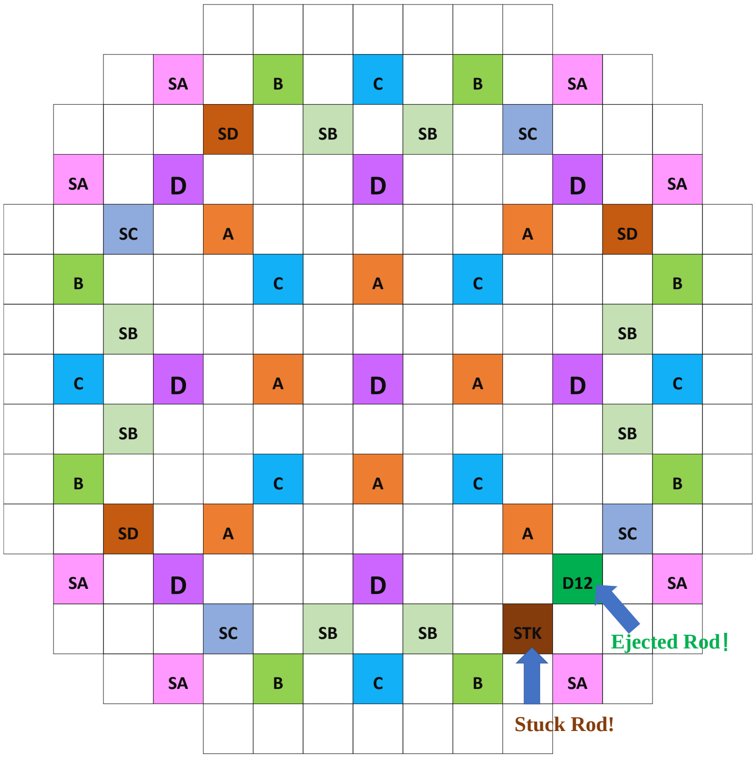


Figure 6.14: Control rod layout of the full-core RIA problem. One RCCA (labeled with D12) of the control bank D is ejected out of the core in 0.1 s. One RCCA (labeled with STK) of control bank SD is stuck. The problem is performed with full symmetry.

large-scale problems. The "hybrid" method is not investigated because the control rods are essentially moving during the whole transient.

Table 6.9: Time step sizes for the full-core HFP RIA problem.

Time Period	Time Step Size	Remark
0.00-0.10 s	5 ms	Rod Ejection
0.10-0.38 s	5 ms	No rod movement
0.38-0.59 s	10 ms	Reactor Trip
0.59-2.79 s	50 ms	Reactor Trip
2.79-3.89 s	100 ms	Reactor Trip
3.89-3.90 s	10 ms	Reactor Trip
3.90-5.00 s	100 ms	Reactor Trip

6.2.4.1 Results

The core power and reactivity results are presented in Figures 6.15 and 6.16, respectively. It can be seen that the three different cases have very similar power histories and reactivity histories.

Figure 6.16 also shows the difference of the reactivities for the TML-3-1G case and TML-4 case compared to the TML-3 case. The maximum differences in reactivities are less than $0.016.

Figure 6.17 compares the core power and maximum linear heat generation rate (linear power) for the TML-3-1G and TML-4 cases with those for the TML-3 case. It can be observed that the relative differences of the core power and maximum linear power share a similar trend, and that the maximum relative differences of the powers for both cases are smaller than 1.2%. Therefore, it can be concluded that using 1G/MGCMFD iteration alone or the TML-4 scheme has comparable accuracy to the current TML-3 scheme with default options. After the SCRAM is triggered and the control rods are inserted to trip the core, the relative differences of the core power and maximum linear power for the TML-3-1G case keep decreasing, while for the TML-4 case, they keep oscillating. This is another indicator that the TML-4 scheme is deficient when the control rods are moving.

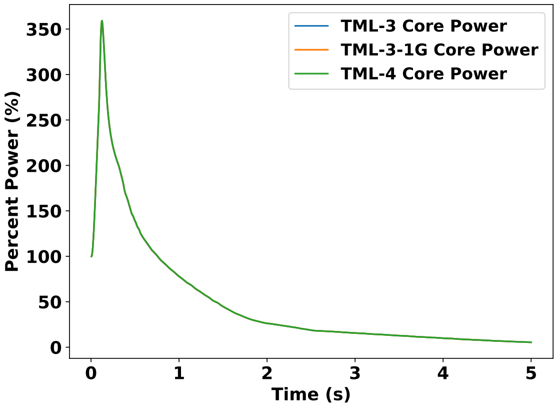


Figure 6.15: Power history of the full-core HFP RIA problem.

The run time results of these three cases are summarized in Table 6.10. There are more abbreviations used in this table. "MR" stands for the run time of the MOC solver; "NR" stands for the run time of the nodal solver; "THR" presents the run time of the TH solver, and "OR" is the run time of other things, including communication, input/output (I/O) reading/writing, etc. It can be seen that both methods presented in this chapter reduce the CMFD time by more than 70%, and the total run time is reduced by more than 41% and at most 49%. The TML-4 case takes more time than the TML-3-1G case. The difference is due to the differences in the run time of the nodal solver caused by the fluctuation in the communication time on the high performance computing (HPC) platform. There are in total 34992 cores on the Falcon HPC cluster. The number of cores used for each job is around one-sixth of the number of all the cores on the cluster. Therefore the randomness in the allocation of the nodes, the possible sharing of the resources with other tasks, the latency in the communication between different nodes, and the I/O reading/writing make the

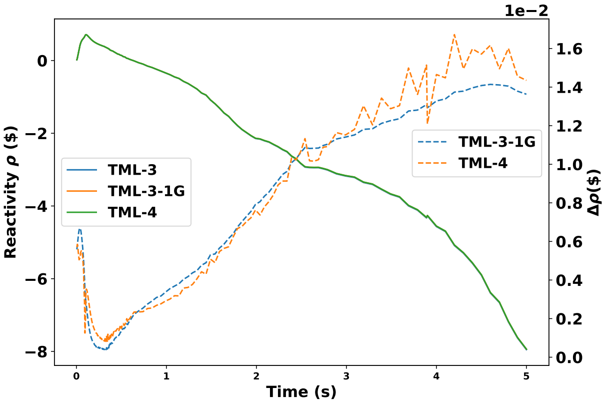


Figure 6.16: Reactivity history and comparison of the full-core HFP RIA problem. The solid lines are reactivity histories and the dashed lines are differences.

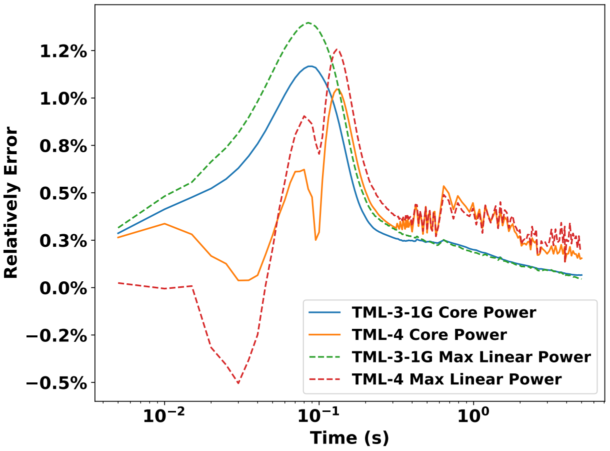


Figure 6.17: Core power and maximum linear rate comparison for the full-core HFP RIA problem. The maximum relative difference is around 1.2% for both methods, indicating the method is applicable for the realistic full-core problems.

run time for the same case inconsistent for different jobs. This conclusion is supported by simulating the same TML-3 case for a second time, with run time (labeled with second trial) presented in Table 6.10. The run time of CMFD solver, MOC sweeper, TH solver, and other solvers for the first trial and the second trial are very close, but the nodal solver's run time is different. The difference in the total run time of these two simulations (that is 5.8 hours) can be directly attributed to the difference in the run time of the nodal solvers (that is 5.1 hours). Moreover, the run time of the nodal solver for the first trial is similar to that for the TML-4 case, while the nodal solver run time for the second trial is similar to that for the TML-3-1G case. Therefore it can still be concluded that the 1G/MGCMFD iteration and TML-4 have comparable efficiency.

k_{eff} results from the steady-state calculation and the adjoint calculation for using multigrid solver (that is used in TML-4 and TML-3-1G) and Krylov solver (that is used in TML-3) are

Table 6.10: Run time for different cases of the full-core HFP RIA problem.

Case	TR (hrs)	CR (hrs)	MR (hrs)	NR (hrs)	THR (hrs)	OR (hrs)	TR RD	CR RD
TML-3 (first trial)	46.7	23.09	11.55	7.24	1.83	2.99	N/A	N/A
TML-3-1G	24.78	6.26	11.42	2.12	1.63	3.4	47%	73%
TML-4	29.15	5.11	11.54	7.07	1.55	3.87	41%	78%
TML-3 (second trial)	40.94	22.63	11.26	2.17	1.72	3.15	N/A	N/A

shown in Table 6.11. Theoretically, the k_{eff} of the forward problem and the adjoint problem are identical. The results however indicate that the difference in the k_{eff} of two problems is 17 pcm using multigrid solver, while is 0.02 pcm using Krylov solver. The difference in the k_{eff} of the two methods in the forward problem is 0.57 pcm. Therefore, the current implementation of the multigrid solver is still problematic for the adjoint calculation and indicates the solution of CMFD TFSP is inaccurate.

Table 6.11: Comparison of k_{eff} for the full-core HFP RIA problem.

Method	Forward k_{eff}	Adjoint k_{eff}	Forward/Adjoint $\|\Delta k\|$ (pcm)
Krylov	0.9932692	0.9932690	0.02
Multigrid	0.9932635	0.9930789	18.46
$\|\Delta k\|$ (pcm)	0.57	19.01	N/A

6.3 Summary

In this chapter, the one group CMFD was introduced to accelerate the TML scheme in the MPACT code. A dynamic 1G/MGCMFD iteration strategy was developed and a new 1GCMFD level was introduced in the TML hierarchy. The new scheme was referred to as TML-4 for the four levels in the phase space solution of the transport transient kinetics problem. The 1G/MGCMFD iteration was shown to be effective by accelerating the convergence of the fission source of the MGCMFD TFSP problem. For TML-4, in the coupling of MGCMFD/1GCMFD, the multigroup flux was directly updated with the one group flux. The underlying assumption for this approach was that the MGCMFD flux could be factorized as the product of a shape function in energy space and a one-group amplitude function, with the shape function in energy space varying more slowly than the one-group flux in time. The new TML-4 scheme with correcting multigroup flux directly had comparable efficiency to using 1G/MGCMFD iteration alone.

In terms of accuracy, it was observed that the error introduced by the 1G/MGCMFD iteration was negligible. However, this method did not improve the accuracy of the TML-3 scheme. For the TML-4 scheme, numerical results show that using TML-4 may not be a good idea when the control rods are moving. However, when the control rods are not moving, compared with using 1G/MGCMFD iteration alone in the TML-3 scheme with the default option, TML-4 is more accurate and efficient. Therefore, this chapter suggests a hybrid method that uses the default TML-3 option with a 1G/MGCMFD iteration when the control rods were moving and uses TML-4 when the control rods are not moving. The numerical results show that the hybrid method is more accurate and has comparable efficiency to using the 1G/MGCMFD iteration alone or using the TML-4 scheme.

The effectiveness of the TML-4 method was demonstrated by solving a series of transient test problems in MPACT ranging from a 4×4 assembly test problem to a full-core practical application. For smaller problems, the 1G/MGCMFD iteration and the new TML-4 scheme could reduce the run time of the CMFD solver by 50%, and the total run time by at least 16%. However, for large-scale full-core problems, the run time of the CMFD solver could be reduced by 78%, and the total run time was reduced by as much as 47%.

The results of this work show that the TML method with 1GCMFD acceleration improved the efficiency of the TML scheme and could also improve the accuracy of some transient processes. This work reduced the computational burden for full-core transport transient solutions and improved the feasibility of using the MPACT code for solving practical transient applications.

CHAPTER 7

Improvements of the Coupled Transient Simulation

The 1GCMFD acceleration in Chapter 6 improves the efficiency of the TML scheme, but does not improve its accuracy. In this chapter, we describe three modifications to make the TML scheme more accurate than the scheme presented in [51, 54, 50, 105]–especially in simulations with feedback.

The chapter is organized as follows. In Section 7.1, we use the Spectral Deferred Correction (SDC) method, a stable and robust, efficient, and almost arbitrary-order accurate time-integration method for the solution of the EPKE problems in TML. In Section 7.2, we modify the approximation of the CMFD coefficients at the beginning of the CMFD level every step and propose a new implementation of the TML-4 scheme. Finally, in Section 7.3, a better coupling scheme of the neutronics and feedback is introduced for multiphysics simulations with adaptive time steps.

7.1 Spectral Deferred Correction Method

A lot of methods have been developed to solve the initial value problem. In the time-dependent neutron transport simulation, most methods used are simple low-order implicit methods. One example is the backward Euler (BE) method, that is the time-integration method for solutions to the governing equations on each level of TML. However, BE is just first-order accurate. Although ultra-fine time steps of 0.1 ms have been used on the EPKE step in the default TML settings, the time step size is still not small enough.

To motivate the use of the high-order accurate method, a simple numerical study to show the limitation of the BE method is performed on a PKE problem with the delayed neutron data shown in Table 7.1. The term PKE rather than EPKE is used here because the parameters of the equations are not obtained from the transport solutions. The PKE problem is simulated with time steps of 0.1 ms for 0.1 s using the BE scheme. The reference solution is calculated analytically. The results are shown in Figure 7.1. It can be seen that the relative error of the power result at the end of the simulation is around 4%, when the initial reactivity is \$1.25, and 16% when reactivity is \$1.5.

Table 7.1: Delayed parameters of a PKE problem to show the limitation of BE method. All the PKE models used in the dissertation uses the parameters listed here unless notified. These parameters are from the lecture notes of NERS 551 in University of Michigan.

Group	Delayed Neutron Yield β	Precursor Decay Constants λ (s^{-1})	Prompt Neutron Generation Time Λ (s)
1	2.584E-4	0.0128	2.57E-5
2	1.5200E-3	0.0318	
3	1.3908E-3	0.119	
4	3.0704E-3	0.3181	
5	1.1020E-3	1.4027	
6	2.584E-4	3.9286	

Here, p is the power. Since the transport solution is eventually corrected by the PKEs solutions, it would be impossible for the pin power results to become more accurate than the PKEs' solution. Therefore, high-order methods that do not exhibit the same behavior as BE should be used.

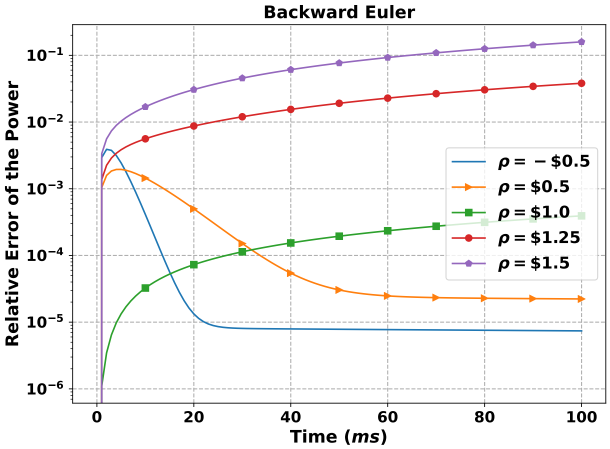


Figure 7.1: Accuracy of BE method in various problems. The PKE problem is simulated for 0.1 s with time steps of 0.1 ms. The relative errors of the power is calculated for each time steps. It can be seen if a reactivity of \$1.5 is inserted into the system, the relative error at the end of simulation can be as large as 16%. The reference solutions are generated analytically.

The most popular high-order methods are the Theta method, Backward Differential Formula (BDF), and implicit Runge-Kutta methods. These methods can be classified as "intrinsic high-order methods" [106]. With the order of accuracy increasing, the implementation of these methods becomes more complicated, and limited by their stability or efficiency. For example, there exists no stable BDF method of order greater than six; and high-order implicit Runge-Kutta method can be extremely computationally expensive [107].

Alternative approaches for these high-order methods are methods based on accelerating the convergence of low-order schemes [106]. A representative of these methods, that is first proposed in [106], is the SDC method. The method is developed from the classical deferred correction method [108], can be arbitrary-order accurate, and is extremely stable for stiff problems, even when its order of accuracy is very high. The most attractive feature of SDC is that the method is constructed based on the low-order problem (BE or forward Euler). The difference between the SDC method and the classical deferred correction method is that spectral integration, that is introduced in Section 7.1.1.2, is used. The method is straightforward to implement, and the order of accuracy can be adjusted readily without altering the implementation. In the field of nuclear engineering, some researchers have introduced SDC or other deferred correction methods to solve the point kinetics equations [109, 110]. Researchers in other fields have introduced the SDC method and its variants to construct high-order stable solvers, and performed stable and efficient high-order simulations for multiphysics problems [111, 112, 113].

In our research, due to time limitations, we only use SDC to solve the PKEs and improve the accuracy of the EPKE solver in MPACT because it is the easiest to fix and also critical for the accuracy of the results. We also develop methods to apply the SDC to the PKE with linear energy feedback (PKE-EF) [88], which is a simple low-order model for the neutronics problem coupled with feedback from TH. Though these problems are simple, the methodologies developed for solving the coupled precursor and feedback equations and PKEs when implementing the SDC do provide some insights. In the future, the investigations and methods developed here should help in the development of new, high-order, stable, and efficient solvers for the neutron transport transient calculations.

7.1.1 Theory

7.1.1.1 Mathematical Preliminaries

The SDC method is developed based on the Picard integral equation. The governing equation of the initial value problem for the range $[a, b]$ is assumed to be:

$$\frac{d\Psi}{dt} = F(t, \Psi)\,,\ t \in [a, b]\,, \tag{7.1}$$

$$\Psi(a) = \Psi_a\,, \tag{7.2}$$

where Ψ_a, $\Psi(t)$ are in $\mathbb{C}^N$, and $F : \mathbb{R} \times \mathbb{C}^N \to \mathbb{C}^N$. N is the dimension and Ψ is the solution we want to obtain. We also assume that $F \in C^\infty(\mathbb{R} \times \mathbb{C}^N)$, i.e. F is infinitely differentiable or sufficiently smooth. Then the solution Ψ is

$$\Psi(t) = \int_a^t F\big(\tau, \Psi(\tau)\big) d\tau + \Psi_a \,. \tag{7.3}$$

It should also be noted that the range $[a, b]$ is typically assumed to a time step.

Now, suppose that we have obtained the approximate solution $\tilde{\Psi}(t)$. The error of the solution denoted by $\delta(t)$ is calculated by:

$$\delta(t) = \Psi(t) - \tilde{\Psi}(t) = \int_a^t F\big(\tau, \Psi(\tau)\big) d\tau + \Psi_a - \tilde{\Psi}(t) \,. \tag{7.4}$$

The definition of error is substituted back into the Eq. (7.4), then we have

$$\delta(t) = \int_a^t F\big(\tau, \tilde{\Psi}(\tau) + \delta(\tau)\big) d\tau + \Psi_a - \tilde{\Psi}(t) \,. \tag{7.5}$$

The residual function [106] as suggested by Eq. (7.6) is also introduced.

$$\epsilon(t) = \int_a^t F\big(\tau, \tilde{\Psi}(\tau)\big) d\tau + \Psi_a - \tilde{\Psi}(t) \,. \tag{7.6}$$

Then we have:

$$\Psi_a - \tilde{\Psi}(t) = \epsilon(t) - \int_a^t F\big(\tau, \tilde{\Psi}(\tau)\big) d\tau \,. \tag{7.7}$$

Substituting Eq. (7.7) into Eq. (7.5), we can obtain:

$$\delta(t) = \int_a^t F\big(\tau, \tilde{\Psi}(\tau) + \delta(\tau)\big) d\tau - \int_a^t F\big(\tau, \tilde{\Psi}(\tau)\big) d\tau + \epsilon(t) \,. \tag{7.8}$$

Next, defining the function $G : \mathbb{R} \times \mathbb{C}^n \to \mathbb{C}^n$ to be:

$$G(t, \delta) = F\big(t, \tilde{\Psi}(t) + \delta(t)\big) - F\big(t, \tilde{\Psi}(t)\big) \,, \tag{7.9}$$

we rewrite Eq. (7.8) in the form:

$$\delta(t) - \int_a^t G(\tau, \delta(\tau)) d\tau = \epsilon(t) \,. \tag{7.10}$$

This is governing equation of the error to correct the solution in the classical deferred correction, and has the similar form to Eq. (7.3).

In the next several sections, how the residual function is calculated and how the error is solved are presented.

7.1.1.2 Spectral Integration

To calculate the residual function in Eq. (7.6), the integral:

$$\int_a^t F\big(\tau, \tilde{\Psi}(\tau)\big) d\tau \,, \tag{7.11}$$

must be evaluated. In SDC, the spectral integration method is used. The derivation process is presented here.

The governing equation must be first solved with multiple sub-steps. Therefore, there is a fine grid inside the time range $[a, b]$. The time points are referred to as nodes in this chapter, following the convention used in [106]. M nodes are used with $a = t_0 < t_1 < t_2 < \cdots t < t_M <= t_{M+1} = b$.

It is also assumed that the approximated solution $\tilde{\Psi}$ has been obtained. The derivatives $F(t_m, \tilde{\Psi}_m)$ are calculated for $m = 1, 2, 3 \cdots M$, and $F(t)$ can be obtained via high-order Langrage interpolation. Then, the residual function can be obtained by integrating $F(t)$ analytically. To illustrate this, let the matrices $\mathbf{F} = \{F_1, \cdots, F_m, \cdots, F_M\}^T$ (F_m is already a vector) and $\mathbf{I} = \{\int_a^{t_1} F d\tau, \int_a^{t_2} F d\tau, \cdots \int_a^{t_M} F d\tau\}^T$. It turns out that

$$\mathbf{I} = \mathbf{SF} \,, \tag{7.12}$$

where $\mathbf{S} \in \mathbb{C}^{M \times M}$, i.e., the process of obtaining $\mathbf{I}$ from $\mathbf{F}$ is a linear mapping [106]. This is the general procedure of the deferred correction method. Since Langrage interpolation is affected by the size of the sub-steps, when $b - a$ changes, the $\mathbf{S}$ will change.

Compared to the standard deferred correction method, the improvement from SDC is that *time nodes for the SDC are obtained spectrally. Therefore Eq.* (7.11) *can be efficiently evaluated via spectral integration [114].*

The matrix $\mathbf{S}$ obtained via spectral integration in Eq. (7.12) is called the spectral integration matrix. Suppose that we have obtained abscissas $\mathbf{x}$ and the weights $\mathbf{w}$ for the Gauss-Legendre quadrature set with M points. The m^{th} internal node t_m is determined by:

$$t_m = \frac{b+a}{2} + \frac{b-a}{2} x_m = \frac{b+a}{2} + \frac{h}{2} x_m \,. \tag{7.13}$$

Here h is the step size, i.e., $b - a$. Let the vector $f \in \mathbb{C}^{(m)}$ be a column vector of F. Then the Lagrange interpolation is written as

$$\hat{f}(t) = \sum_{l=0}^{M-1} a_l \hat{P}_l(t) = \sum_{l=0}^{M-1} a_l P_l\bigg(\frac{2t - (a+b)}{h}\bigg) \,, \tag{7.14}$$

where P_l is the Legendre polynomial, and

$$\hat{P}_l(t) = P_l\biggl(\frac{2t-(a+b)}{h}\biggr)\,. \tag{7.15}$$

The coefficients a_l are

$$\begin{aligned} a_l &= \frac{2l+1}{2}\int_{-1}^{1} \hat{f}\biggl(\frac{h}{2}\tau + \frac{a+b}{2}\biggr)P_l(\tau)d\tau \\ &= \frac{2l+1}{2}\sum_{m=1}^{M} w_m f_m P_l(x_m) = V_l^T f\,, \end{aligned} \tag{7.16}$$

where V_l the column vector

$$V_l = \frac{2l+1}{2}[w_1P_l(x_1), w_2P_l(x_2), \cdots w_MP_l(x_M)]^T \in \mathbb{C}^{(m)}\,. \tag{7.17}$$

Now, let the matrix

$$\mathbf{V} = \{V_0^T, V_1^T, \cdots V_{M-1}^T\}\,. \tag{7.18}$$

Substituting Eq. (7.16) into Eq. (7.14), and letting the vector $\hat{P}(t) = \{\hat{P}_0(t), \hat{P}_1(t)\cdots \hat{P}_{M-1}(t)\}$, we have

$$\hat{f}(t) = \hat{P}(t)\mathbf{V}^T f\,. \tag{7.19}$$

Following this, we introduce the matrix $\overline{\mathbf{V}} \in \mathbf{C}^{M\times M}$. The component $\overline{v}_{l+1,m}$ is defined via the integral:

$$\begin{aligned} \int_a^{t_m} \hat{P}_l(\tau)d\tau &= \frac{h}{2}\int_{-1}^{x_m} P_l(s)ds = \frac{x_m+1}{2}\frac{h}{2}\int_{-1}^{1} P_l(s')ds' \\ &= \frac{x_m+1}{2}\frac{h}{2}\sum_{i=1}^{M} P_l\bigl((x_m+1)(x_i+1)-1\bigr)w_i = \frac{h}{2}\overline{v}_{l+1,m}\,. \end{aligned} \tag{7.20}$$

The integration term, i_m now becomes

$$i_m = \int_a^{t_m} \hat{f}(\tau)d\tau = \frac{h}{2}\overline{\mathbf{V}}[:,m]\mathbf{V}^T f\,. \tag{7.21}$$

Next, defining the column vector $\mathbf{i} = \{i_1, i_2 \cdots i_M\}^T$ gives

$$\mathbf{i} = \frac{h}{2}\overline{\mathbf{V}}\mathbf{V}^T\,. \tag{7.22}$$

We denote the $M\times M$ matrix $\mathbf{S}_b$ by:

$$\mathbf{S}_b = \overline{\mathbf{V}}\mathbf{V}^T\,. \tag{7.23}$$

$\mathbf{S}_b$ only depends on M, thus it can be precomputed. Now it should be straightforward to see that the integration matrix

$$\mathbf{S} = \frac{h}{2}\mathbf{S}_b \,. \tag{7.24}$$

Then Eq. (7.12) is written by

$$\mathbf{I} = \frac{h}{2}\mathbf{S}_b\mathbf{F} \,. \tag{7.25}$$

[107] provides another derivation for the spectral integration matrix involved with matrix inversion. The process presented here provides an alternative way to calculate the spectral integration matrix directly.

Other popular applicable spectral quadrature sets are Chebyshev, and Gauss-Lobatto quadrature sets [107, 113]. The quadrature sets affect the derivation of spectral integration and slightly affect the stability region of the SDC. Discussions on the quadrature sets can be found in [107]. We proceed using the Gauss-Legendre quadrature sets because it is the most popular and commonly used in the nuclear reactor physics field.

Using the correction, the $(M-1)^{th}$-order approximation of the residual function at the time nodes can be written as

$$\sigma(\overline{\mathbf{\Psi}}) = \mathbf{SF} - \overline{\mathbf{\Psi}}_a - \overline{\mathbf{\Psi}} \,, \tag{7.26}$$

with

$$\begin{aligned} \overline{\mathbf{\Psi}} &= \{\tilde{\Psi}(t_1), \tilde{\Psi}(t_2), \cdots, \tilde{\Psi}(t_M)\}^T \,, \\ \overline{\mathbf{\Psi}}_a &= \{\tilde{\Psi}(a), \tilde{\Psi}(a), \cdots, \tilde{\Psi}(a)\}^T \,, \\ \sigma, &\overline{\mathbf{\Psi}}, \overline{\mathbf{\Psi}}_a \in \mathbb{C}^{M\times N} \,. \end{aligned} \tag{7.27}$$

We use the same notation σ as that in [106] to indicate that σ is a high-order approximation to the residual ϵ.

7.1.1.3 Error Correction

The error to correct the solution is obtained by solving Eq. (7.10) with the low-order method. It is recommended that forward Euler (for non-stiff problems) or BE method (for stiff problems) should be used [106].

The forward Euler method for the solution of Eq. (7.10) in the non-stiff problem is given by the formula:

$$\delta_{m+1} = \delta_m + G(t_m, \delta_m)h_{m+1} + \sigma_{m+1} - \sigma_m \,, \tag{7.28}$$

and the BE method for the stiff problem is:

$$\delta_{m+1} = \delta_m + G(t_{m+1}, \delta_{m+1})h_{m+1} + \sigma_{m+1} - \sigma_m \,, \tag{7.29}$$

for $m = 1, 2, \cdots M$, and $h_{m+1} = t_{m+1} - t_m$. Sometimes, G can be partitioned as:

$$G = G_I + G_E, \tag{7.30}$$

where G_I is stiff and can be solved implicitly, and G_E is non-stiff and can be solved explicitly. Then it is also possible to solve Eq. (7.10) in the implicit-explicit (IMEX) way [111]:

$$\delta_{m+1} = \delta_m + G_I(t_{m+1}, \delta_{m+1})h_{m+1} + G_E(t_m, \delta_m)h_{m+1} + \sigma_{m+1} - \sigma_m\,. \tag{7.31}$$

The Spectral Deferred Correction for the solution of Eq. (7.1) is a prediction-correction process. During the *prediction*, the solution is computed with a low-order numerical method at spectral time nodes $t_0, t_1, \cdots, t_{M+1}$. The solution is denoted as $\overline{\mathbf{\Psi}}^{(0)}$. During the *correction*, there are J correction "sweeps". For the j^{th} correction, the residual σ^j is calculated with $\overline{\mathbf{\Psi}}^{(j)}$ using Eq. (7.26). The error can then be calculated with Eqs. (7.28), (7.29) and (7.31), and is used to correct the solution $\overline{\mathbf{\Psi}}^{(j-1)}$, and the updated solution is $\overline{\mathbf{\Psi}}^{(j)}$.

The algorithm of the SDC method is listed in Algorithm 13. J is the number of corrections.

Algorithm 13 General Procedure for Spectral Deferred Correction.

```
Input the number of corrections; initial solution at time $a$, i.e., $\Psi_a$.
*Prediction*: Compute the initial estimation of $\overline{\mathbf{\Psi}}^{(0)}$ at the time nodes $t_1 \cdots t_M$ with the lower-order solver.
*Correction*:
for $j = 1, 2, \ldots, J$ do
    Compute the residual function $\sigma$ at each node via spectral integration.
    Compute the error $\delta$ via the low-order solvers.
    Update the solution for the $j^{th}$ correction by $\overline{\mathbf{\Psi}}^{(j)}(t_m) = \overline{\mathbf{\Psi}}^{(j-1)}(t_m) + \delta(t_m)$.
end for
```

7.1.1.4 Right Endpoint Evaluation

It should be noted that the error is not obtained for time point b as indicated by Algorithm 13. Suppose that J corrections are used. There are multiple ways to calculate the error or the solution at b. Some examples are listed here.

- **Extrapolation** (EP): The solution at b is not calculated for both the prediction and correction. After J corrections, the solution $\Psi(b)$ is then obtained by the Lagrange interpolation [106, 107]. It is also possible to use the spectral integration:

$$\Psi(b) = \left(\overline{\mathbf{\Psi}}^{(J)}\right)^T \mathbf{V1}\,, \tag{7.32}$$

$\mathbf{1} \in \mathbb{C}^{(m)}$ is a column vector with all the components 1.

- **Right-hand Approach** [107]: The time step size $h = \frac{2}{1+x_M}(a-b)$. Then $t_M = b$. The $\Psi(b) = \Psi^{(J)}(t_m)$. This approach is not investigated in here.
- **Gaussian Integration** (GI): The residual at b is calculated via numerical integration using Gauss quadrature:
$$\sigma_{M+1} = \frac{h}{2}\mathbf{F}\mathbf{w}\,. \tag{7.33}$$
Then the error at b is solved with the residual. So far, we have not found any researchers report this treatment. However, it is a straightforward implementation. It is possible that other researchers have already adopted this method.

Some researchers also combine the correction process with solution process together; the details can be found in [111]. Other groups have investigated using the trapezoidal method in the prediction and correction steps [115]. The improvement in this case is small because the time nodes are not equispaced and the resulted SDC is not L-stable.

7.1.1.5 Order of Accuracy and Stability Region

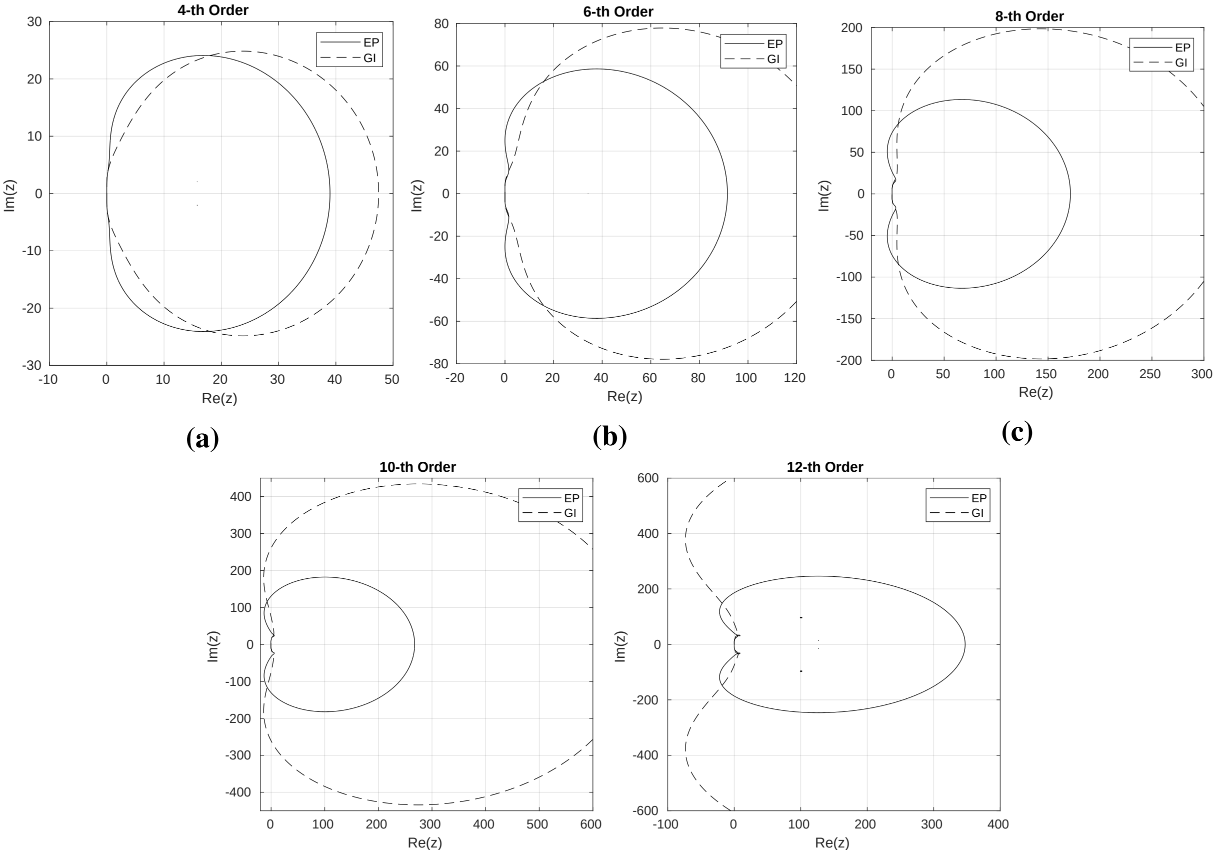
Theoretical proof in [106, 115] and results in [107] show that when F is sufficiently smooth, the solution computed by SDC converges to the exact solution with order $\min(M, J+1)$ when the Euler solver is used. Numerical results later shown in this section show that when GI is used for calculating the residual at the endpoint, the order of accuracy is $\min(2M, J+1)$. The same maximum order of accuracy has been observed in [116, 117].

The A-stability investigation of the SDC method is performed for the problem:

$$\begin{aligned} y' &= -\lambda y, \lambda < 0\,, \\ y(0) &= 1\,. \end{aligned} \tag{7.34}$$

In this analysis, suppose that the method performs for a single step with the step size h. Defining $z = \lambda h$, the value of y at the next step is a function of z, and is denoted as $g(z)$. If $|g(z)| < 1$ for $Re(z) < 0$, then the method is A-stable. If $|g(z)| < 1$ for $\pi - \alpha < arg(z) < \pi + \alpha$, the method is A-(α) stable. It is well known that backward Euler is stable, but more specifically, this notation of “stability” is the A-stability.

The stability regions of SDC based on BE with EP and GI are shown in Figures 7.2. They are the regions outside the lines. It can be seen that SDC is A-(α) stable for all the orders of accuracy investigated here. The approach to evaluate the endpoint value is also observed to affect the stability region. Using GI to calculate the endpoint value can make SDC A-stable for orders up to 8. The results presented here verify that SDC can be stable for very large orders. It turns out the SDC method can be A-(α) stable for orders up to 20 and more details about the stability of SDC can be found in [106, 107].



(a) **(b)** **(c)**

(d) **(e)**

Figure 7.2: Stability region of SDC. $|g(z)| < 1$ for z outside the lines. All the methods are A-(α) stable. For SDC with GI for the endpoint evaluation, it can be A-stable up to 8^{th}-order. $J = M-1$, so M is the order of accuracy.

7.1.2 SDC for PKE

Procedures mentioned in Section 7.1.1.3 provide a skeleton for implementation of the SDC method. The actual implementation of the SDC method depends on the low-order solver. Here we provide implementation details of the SDC solver for PKE.

7.1.2.1 Low-order Solver

In this part, we introduce three different implementations of the low-order solvers for the solution of the prediction step and error of the correction sweeps in PKEs.

The PKEs is written in operator notation:

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{E}(t)\mathbf{P}(t)\,. \tag{2.69 revisited}$$

The first low-order method is referred to as **matrix inversion** (MI). Suppose that the PKEs are solved implicitly. During the prediction step,

$$\mathbf{P}_{m+1} = (I - \mathbf{E}h_{m+1})\mathbf{P}_m\,. \tag{7.35}$$

For the correction, since the operator $\mathbf{E}$ in Eq. (2.69) is linear, $G(t_m, \delta_m)$ may be written as:

$$G(t_m, \delta_m) = \mathbf{E}(t_m)\delta_m\,. \tag{7.36}$$

The BE solution of Eq. (7.29) is:

$$\delta_{m+1} = (\mathbf{I} - \mathbf{E}h_{m+1})^{-1}(\delta_m + \sigma_{m+1} - \sigma_m)\,. \tag{7.37}$$

This is a direct solution process involving inverting a matrix inversion.

The second solver is the **first-order precursor integration (FP)** solver to avoid the matrix inversion [54]. The solution process for the prediction step has been introduced in Section 2.4.3. For the correction step, the FP method will make the low-order solver for the error slightly complicated. The differential form of Eq. (7.10) is:

$$\frac{d\delta(t)}{dt} = \mathbf{E}\delta(t) + \frac{d\epsilon(t)}{dt}\,. \tag{7.38}$$

The error of delayed neutron precursors $\delta(\xi_k)$ can be approximated with

$$\frac{d}{dt}\left(\delta(\xi_k)e^{\lambda_k t}\right) = e^{\lambda_k t}\frac{d\delta(\xi_k)}{dt} + \lambda_k e^{\lambda_k t}\delta(\xi_k) = \frac{\Lambda_0}{\Lambda(t)}\beta_k(t)\delta(p)e^{\lambda_k t} + \frac{d\epsilon_k(t)}{dt}e^{\lambda_k t}\,, \tag{7.39}$$

and the first-order approximation in time is then applied to the residual as:

$$\frac{d\epsilon_k(t)}{dt} = \frac{\sigma_k^{(m+1)} - \sigma_k^{(m)}}{h^{(m+1)}}\,. \tag{7.40}$$

In this paragraph, we use superscript $^{(m)}$ to indicate the variable at time t_m, $\sigma_k^{(m)}$ to indicate the residual of the $\xi_k(t_m)$, and $\sigma_p^{(m)}$ for the residual of $p(t_m)$. Then we have

$$\delta^{(m+1)}(\xi_k) = \delta^{(m)}(\xi_k)e^{-\tilde{\lambda}_k^{(m+1)}} + \Lambda_0\Bigg[\frac{\beta_k^{(m)}\delta^{(m)}(p)}{\Lambda^{(m)}}\frac{\kappa_0(\tilde{\lambda}_k^{(m+1)}) - \kappa_1(\tilde{\lambda}_k^{(m+1)})}{\lambda_k^{(m+1)}}$$

$$
\begin{aligned}
&+ \frac{\beta_k^{(m+1)}\delta^{(m+1)}(p)}{\Lambda^{(m+1)}}\frac{\kappa_1(\tilde{\lambda}_k^{(m+1)})}{\lambda_k^{(m+1)}}\Bigg] + (\sigma_k^{(m+1)} - \sigma_k^{(m)})\frac{\kappa_0(\tilde{\lambda}_k^{(m+1)})}{\lambda_k^{(m+1)}}\,, && (7.41a)\\
\delta^{(m+1)}(p) &= \delta^{(m)}(p) + \left[\frac{\rho^{(m+1)} - \beta^{(m+1)}}{\Lambda^{(m+1)}}\delta^{(m+1)}(p) + \frac{1}{\Lambda_0}\sum_{k=1}^{K}\lambda_k^{(m+1)}\delta^{(m+1)}(\xi_k)\right] \\
&+ \sigma_p^{(m+1)} - \sigma_p^{(m)}\,, && (7.41b)
\end{aligned}
$$

where $\tilde{\lambda}_k^{(m+1)} = \lambda_k^{(m+1)}h^{(m+1)}$. Defining

$$
\begin{aligned}
\hat{S}_k^{(m+1)}(\delta) &= \Bigg[\delta^{(m)}(\xi_k)e^{-\tilde{\lambda}_k^{(m+1)}} + \frac{\Lambda_0\beta_k^{(m)}\delta^{(m)}(p)}{\Lambda^{(m)}}\frac{\kappa_0(\tilde{\lambda}_k^{(m+1)}) - \kappa_1(\tilde{\lambda}_k^{(m+1)})}{\lambda_k^{(m+1)}} \\
&+ (\sigma_k^{(m+1)} - \sigma_k^{(m)})\frac{\kappa_0(\tilde{\lambda}_k^{(m+1)})}{\lambda_k^{(m+1)}}\Bigg]\,, && (7.42a)\\
\tilde{S}^{(m+1)}(\delta) &= \frac{1}{\Lambda_0}\sum_{k=1}^{K}\lambda_k^{(m+1)}\hat{S}_k^{(m+1)}(\delta)\,, && (7.42b)
\end{aligned}
$$

and substituting Eqs. (7.42) back into Eqs. (7.41), yields

$$
\begin{aligned}
\delta^{(m+1)}(p) &= \frac{\delta^{(m)}(p) + h^{(m+1)}\tilde{S}^{(m+1)}(\delta) + \sigma_p^{(m+1)} - \sigma_p^{(m)}}{1 - \tau^{(m+1)}h^{(m+1)} - \frac{\rho^{(m+1)} - \beta^{(m+1)}}{\Lambda^{(m+1)}}h^{(m+1)}}\,, && (7.43a)\\
\delta^{(m+1)}(\xi_k) &= \hat{S}_k^{(m+1)} + \Omega_k^{(m+1)}\delta^{(m+1)}(p)\,,\ k = 1,\cdots K\,. && (7.43b)
\end{aligned}
$$

The variables not introduced here can be found in Section 2.4.3.

The third approach is based on the partitioned approach [111], and is referred to as the **operator splitting (OS)** approach here. In this variation, the prediction step is:

$$
\begin{aligned}
p^{(m+1)} &= \frac{p^{(m)} + \frac{1}{\Lambda_0}h^{(m+1)}\sum_{k=1}^{K}\lambda_k^{(m+1)}\xi_k^{(m)}}{1 - \frac{\rho^{(m+1)} - \beta^{(m+1)}}{\Lambda^{(m+1)}}h^{(m+1)}}\,, \\
\xi_k^{(m+1)} &= \frac{\xi_k^{(m)} + h^{(m+1)}\beta_k^{(m+1)}\frac{\Lambda_0}{\Lambda^{(m)}}p^{(m+1)}}{1 + \lambda_k^{(m+1)}h^{(m+1)}}\,,\ k = 1,\cdots K\,.
\end{aligned}
\qquad (7.44)
$$

For the correction sweeps, the following are used:

$$
\begin{aligned}
\delta^{(m+1)}(p) &= \frac{\delta^{(m)}(p) + h^{(m+1)}\sum_{k=1}^{K}\lambda_k^{(m+1)}\delta^{(m)}(\xi_k) + \sigma_p^{(m+1)} - \sigma_p^{(m)}}{1 - \frac{\rho^{(m+1)} - \beta^{(m+1)}}{\Lambda^{(m+1)}}h^{(m+1)}}\,, \\
\delta^{(m+1)}(\xi_k) &= \frac{\delta^{(m)}(\epsilon_k) + h^{(m+1)}\beta_k^{(m+1)}\frac{\Lambda_0}{\Lambda^{(m)}}\delta^{(m+1)}(p) + \sigma_k^{(m+1)} - \sigma_k^{(m)}}{1 + \lambda_k^{(m+1)}h^{(m+1)}}\,,\ k = 1,\cdots K\,.
\end{aligned}
\qquad (7.45)
$$

The matrix inversion is avoided through separating the operator applied on p and ξ. The OS approach can also be understood as

$$\mathbf{P}^{(m+1)} = \mathbf{P}^{(m)} + (\mathbf{E}_I^{(m+1)}\mathbf{P}^{(m+1)} + \mathbf{E}_E^{(m)}\mathbf{P}^{(m)})h_{m+1}\,, \tag{7.46}$$

where

$$\mathbf{E}_I(t) = \begin{pmatrix} \frac{\rho(t)-\beta(t)}{\Lambda(t)} & & & & & \\ & -\lambda_1(t) & & & & \frac{\Lambda_0}{\Lambda(t)}\beta_1 \\ & & -\lambda_2(t) & & & \frac{\Lambda_0}{\Lambda(t)}\beta_2 \\ & & & \ddots & & \vdots \\ & & & \cdots & -\lambda_K(t) & \frac{\Lambda_0}{\Lambda(t)}\beta_K(t) \end{pmatrix}, \tag{7.47}$$

and

$$\mathbf{E}_E(t) = \begin{pmatrix} 0 & \frac{1}{\Lambda_0}\boldsymbol{\lambda}(t) \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \tag{7.48}$$

with $\boldsymbol{\lambda} = \{\lambda_1(t), \lambda_2(t), \cdots, \lambda_K(t)\}$. $\mathbf{E_I P}$ is the implicit part and $\mathbf{E_E P}$ is the explicit part.

7.1.2.2 Algorithm

The algorithm of the SDC method is listed in Algorithm 14. J is the number of corrections.

Algorithm 14 Spectral Deferred Correction for PKEs.

```
Prediction: Compute the initial estimation of $\mathbf{P}^{(0)}$ at the time nodes $t_0, t_1 \cdots t_{M+1}$ with the lower-order solver in Section 7.1.2.1.
Correction:
for $j = 1, 2, \ldots, J$ do
    Compute the residual function $\sigma$ at each node via spectral integration. Gaussian integration is used if the residual function at the right endpoint is needed.
    Compute the error $\delta$ via the low-order solvers in Section 7.1.2.1.
    Update the solution for the $j^{th}$ correction by $\mathbf{P}^{(j)}(t_m) = \mathbf{P}^{(j-1)}(t_m) + \delta(t_m)$.
end for
```

7.1.2.3 Numerical Results

The parameters from Table 7.1 are used for verifying the accuracy of the SDC method in solving the PKEs. The test code is implemented in MATLAB. The Gauss-Legendre quadrature sets with 4 time nodes are used. Figure 7.3 shows the effect of the treatment of the solution at the endpoint. The dashed line represents of the plot for fitted order of accuracy. When ED is used, the maximum order of accuracy observed is M. When the Gaussian integration is used, the maximum order of accuracy observed is $\sim 2M$. FP is used as the low-order PKE solver, because it is the default low-order solver for the EPKEs in MPACT.

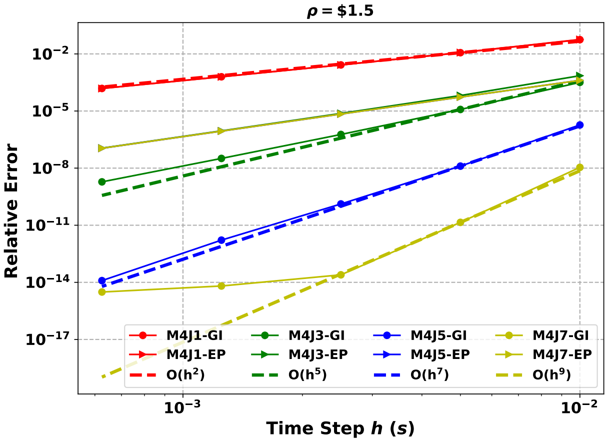
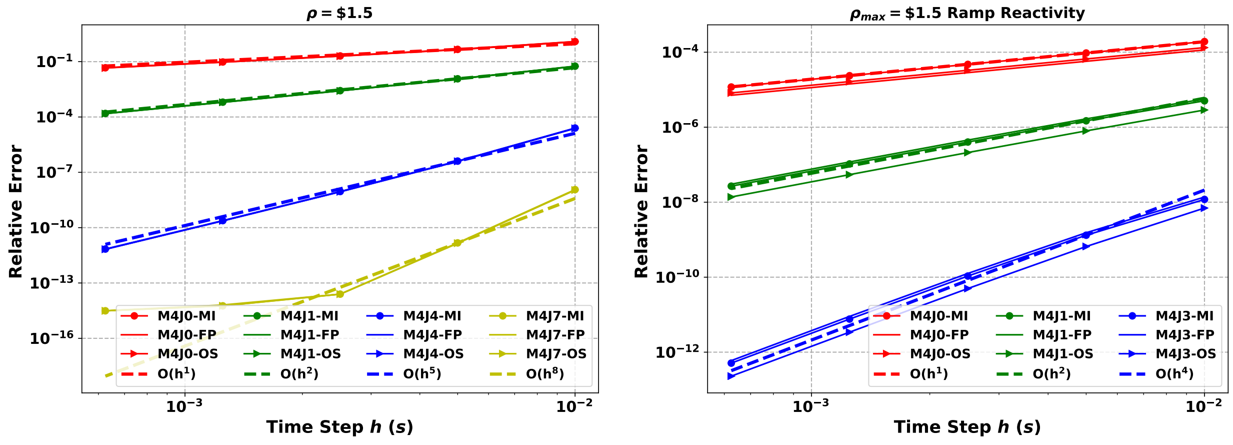


Figure 7.3: Effect of the endpoint calculation on order of convergence. A PKE problem with a step reactivity insertion of \$1.5 is simulated for 0.4 s. “GI” stands for Gaussian integration, while “EP” denotes the extrapolation.



(a)

(b)

Figure 7.4: Effect of low-order solvers for SDC on order of convergence. (a) simulates the PKE problem with a step reactivity insertion of \$1.5 for 0.4 s. The reactivity simulated in (b) increases linearly from 0 to \$1.5 in 0.4 s. Low-order solvers have a minor effect on the accuracy of SDC, but do not affect the overall order of accuracy. The dashed line is the expected order of accuracy. The dashed line is the plot of the expected order of accuracy.

Figures 7.4 show how the low-order solvers in Section 7.1.2.1 affect the performance of the SDC methods for the PKEs. Slight differences in the relative error can be observed. The order in which the error converges does not depend on the low-order solvers investigated here. The order

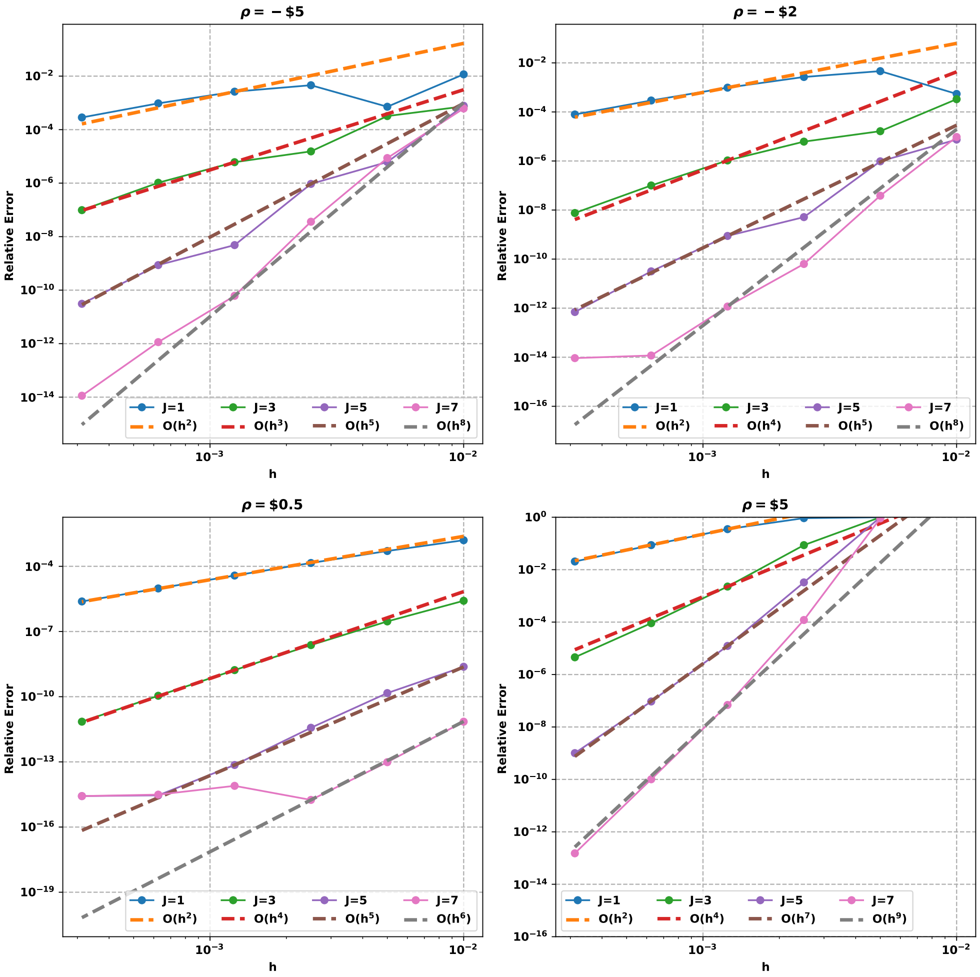


Figure 7.5: Order of convergence of the SDC method for the PKEs with different reactivities. Gaussian integration is used for the residual at the right endpoint. M is 4. The dashed line is the plot of the expected order of accuracy. It can be seen that the observed rate of convergence is around $J+1$.

of accuracy observed here is $\min(2M, J+1)$. No instability is observed for the 8^{th}-order accurate implementation. The results show that SDC is a powerful approach to turn the low-order solver into a high-order solver without losing stability. Currently, the FP or SP is used to approximate the fission source in transient problems to fully couple the flux/power to the precursors [51]. The results here indicate that in the near future, the time-dependent NTE can be developed based on the partitioned SDC method, and the complexity from precursor integration can be avoided.

The SDC method is eventually verified in PKE problems with different reactivities. The results are shown in Figure 7.5. It can be seen that the desired order of convergence is observed. Further, using 1 ms step size for SDC does achieve reasonably accurate results.

After the accuracy of the SDC solver is verified, the SDC method is implemented in MPACT to solve the EPKEs. We adopt the Gaussian integration for calculating the residual at the endpoint of the correction step and FP as the low-order solver. The results of the existing MPACT implementation with the BE solver are compared to the results of the SDC solver. The time step size of the SDC solver is 1 ms with $M = 10$, $J = 4$. $J = 4$ is used because the SDC solver is 5^{th}-order accurate in this case, and the results from Figures 7.4 show a maximum relative error of 10^{-10} that is sufficiently accurate. The results from MPACT are shown in Figures 7.6. Refining the time step size for the BE solver to 1 μs gives the results that are comparable to the SDC results. Therefore, the implementation of SDC in MPACT is verified. When the default 0.1 ms is used for the solution of a pin-cell transient problem with a maximum reactivity of \$1.98 using BE, it can be seen that the current BE solver with the default time step size has a maximum relative error of about 6%.

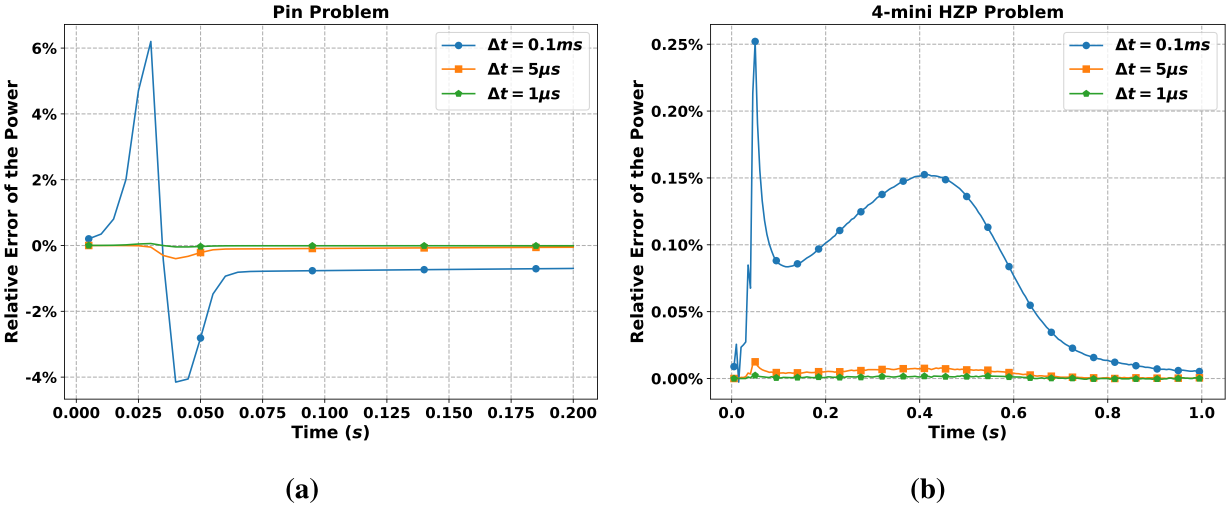


Figure 7.6: Comparison of results for EPKE solvers based on BE to SDC reference in MPACT. Δt is the time step size used for the BE solver. (a) is a pin transient problem with a maximum reactivity of \$1.98. (b) is the 4-mini HZP problem.

Since SDC is much more accurate than BE for the EPKEs, the SDC solver is used for all the results presented later in this dissertation.

7.1.3 SDC for PKE-EF

The SDC method is also used to solve the PKE-EF problem. The PKE-EF is a reduced-order model of the transient problem coupled to TH. The model is used to improve the coupling scheme shown later in this chapter, and to develop the adaptive time-stepping scheme in Chapter 8. The

PKE-EF model [88] can be written as:

$$\frac{dp(t)}{dt} = \frac{\rho(t) - \beta(t)}{\Lambda(t)} p(t) + \frac{1}{\Lambda_0} \sum_{k=1}^{K} \lambda_k(t)\xi_k(t)\,, \tag{7.49}$$

$$\frac{d\xi_k(t)}{dt} = \frac{\Lambda_0}{\Lambda(t)} \beta_k(t)p(t) - \lambda_k(t)\xi_k(t),\ k = 1, 2, \cdots K\,, \tag{7.50}$$

$$\rho(t) = \rho^{ex}(t) + \gamma_d Q(t)\,, \tag{7.51}$$

$$\frac{dQ(t)}{dt} = -\lambda_H Q + p(t) - p(0)\,, \tag{7.52}$$

where λ_H is the single time constant for heat deposition and

$$\lambda_H \approx \frac{1}{2\ s}\,. \tag{7.53}$$

γ_d is the feedback coefficient and

$$\gamma_d \approx -0.8\ \$/fp \cdot s\,, \tag{7.54}$$

is typical for "FBR and LWR oxide-fueled power reactors" [88]. ρ^{ex} is the external reactivity.

We introduce the Implicit-Explicit (IMEX) treatment, i.e.

$$\mathbf{F}_E(t, \mathbf{Y}) = \begin{pmatrix} p(t) - p(0) \\ \mathbf{0} \end{pmatrix}\ ,\ \ \mathbf{F}_I(t, \mathbf{Y}) = \begin{pmatrix} -\lambda_H Q(t) \\ \mathbf{E}(t, Q)\mathbf{P}(t) \end{pmatrix}\ , \tag{7.55}$$

as the low-order solver. The solution is $\mathbf{Y} = \{Q(t), \mathbf{P}^T\}^T$. Then the equation to advance the time step for the IMEX method is:

$$\mathbf{Y}_{m+1} = \mathbf{Y}_m + \mathbf{F}_E(t_m, \mathbf{Y}_m)h_m + \mathbf{F}_I(t_{m+1}, \mathbf{Y}_{m+1})h_m\,. \tag{7.56}$$

The algorithm for solving the PKE-EF problem is similar to that shown in Algorithm 14 and is omitted here. For a problem with ρ^{ex} = \$1.5, λ_H = 0.5/s, $\gamma_d = -0.8$ \$/fp·s and the initial condition $p(0) = 10^{-4}$ fp, the effect of the number of corrections is shown in Figure 7.7. It can be seen that the order of the accuracy is again $\sim J + 1$, and agrees with the theoretical predictions in [106]. Since the problem is solved with SDC method based on IMEX treatment, the SDC method is denoted as SDC-IMEX. It should be noted that this problem is nonlinear since ρ is a function of Q. The results imply that it is possible to develop SDC for simulating the time-dependent neutron transport problem with multiphysics feedback.

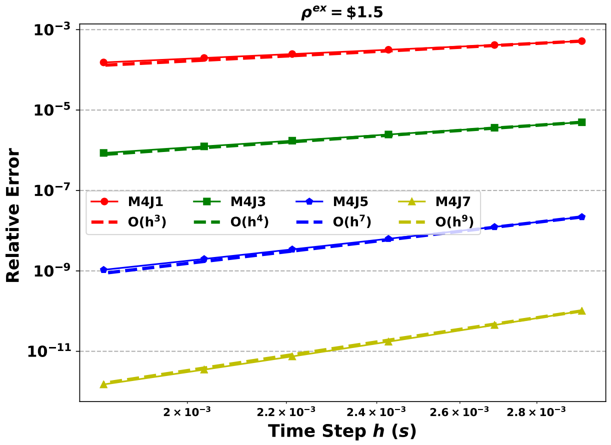


Figure 7.7: Convergence study of a PKE-EF solver based on SDC. The dashed line is the plot of the expected order of accuracy. It can be seen that the order of accuracy is around $J + 1$.

7.2 Improvement of CMFD Level

7.2.1 Decoupling TH inside the TML

As has been presented in Chapter 6, the TML scheme comprises three levels of calculation: the transport level, the CMFD level, and the EPKE level. When feedback from TH is considered in the simulation, the time-dependent neutron transport equation is solved using operator splitting.

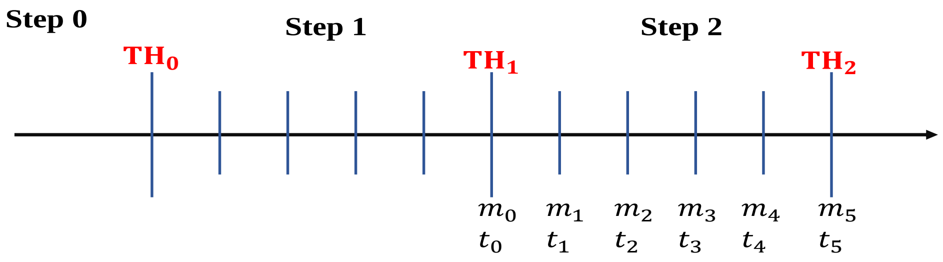


Figure 7.8: Illustration of TH coupling for TML. This figure is reproduced from Fig.2 in [54].

Because of how the CMFD coefficients are constructed in the TML, the feedback from the TH is implicitly incorporated. As shown in Figure 7.8, steps 0, 1 and 2 represent the transport steps, and each transport step is subdivided into several fine CMFD steps. For step 2, the CMFD coefficients are linearly interpolated with time evolving from t_0 to t_5, i.e., the CMFD coefficients at intermediate CMFD steps are calculated by:

$$\Sigma_x(t_i) = \Sigma_x(t_0)\eta + \Sigma_x(t_5)(1-\eta),\ \eta = \frac{t_i - t_0}{t_5 - t_0}\,. \tag{7.57}$$

For the original TML implementation, the left endpoint CMFD coefficients, $\Sigma_x(t_0)$, are obtained from homogenization with the fine mesh scalar flux at t_5, i.e.:

$$\Sigma_x(t_0) = \Sigma_x\big(TH_1, m_0, \psi(TH_1, t_5, m_5)\big) \,. \tag{7.58}$$

Since the flux at t_5 is the solution *after feedback is applied*, the treatment of Eq. (7.58) takes the feedback implicitly into account as well. This is **approximation 3** in [54].

However, after more experience in simulation cases simulated with the TML scheme, it was observed that approximation 3 did not perform as well as had been expected. The method could induce significant errors in some cases. Furthermore, it was found that another approximation, which determines the CMFD coefficients at t_0 by:

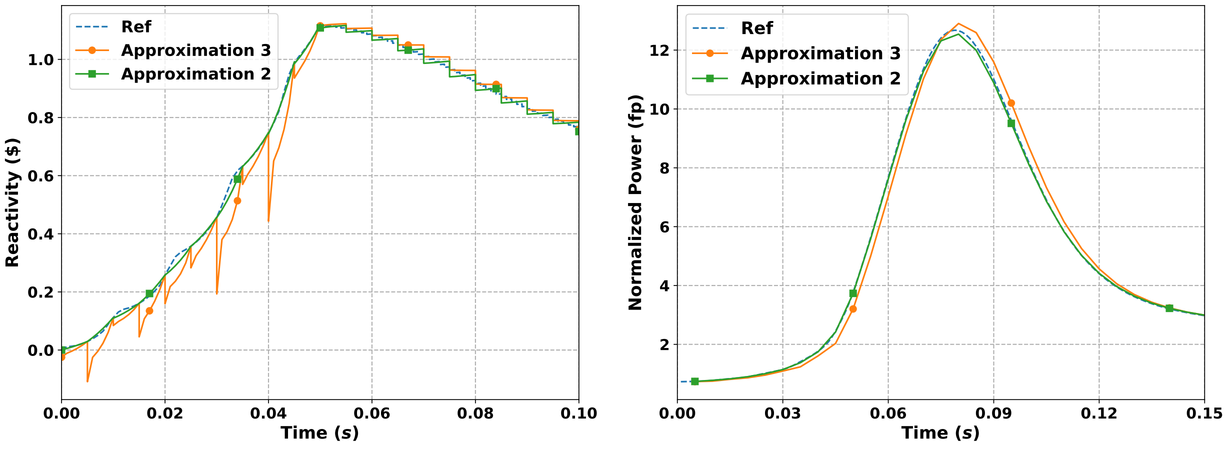
$$\Sigma_x(t_0) = \Sigma_x\big(TH_1, m_0, \psi(TH_0, t_0, m_0)\big) \,, \tag{7.59}$$

predicted more accurate results. This treatment is **approximation 2** in [54].

A severe deficiency observed in approximation 3 is that it *underestimates the reactivity when the control rod is withdrawn from the core, and overestimates the reactivity when the control rod is inserted into the core*. The 4-mini HFP test problem is used as an example with results shown in Figures 7.9. The reference solution is generated with time steps of 1 ms. It can be seen that the reactivity history and the power history of the approximation are underestimated. The maximum relative error is 17%. For each transport step, a "dip" shows up on the reactivity profile at the starting point of the CMFD level. Compared to approximation 3, approximation 2 performs much better.

We use Figures 7.10 to explain the reason. Assume that the control rod tip is at h_b at t_0 and is withdrawn to h_a at t_5. Then the flux at $\psi(h_b, t_0) << \psi(h_b, t_5)$. Therefore, using flux $\psi(h_b, t_5)$ to calculate the CMFD coefficients at t_0 overestimates the absorption cross section, and underestimates the reactivity. Conversely, if the control rod tip is at h_a initially and inserted to h_b at t_5, then flux $\psi(h_a, t_0) >> \psi(h_a, t_5)$. Therefore, the CMFD absorption coefficients will be underestimated, and the reactivity overestimated. Since the flux is eventually corrected by the solution of the EPKEs, for rod insertion and withdrawal, the power will be overestimated and underestimated, respectively.

The comparison of approximation 2 to approximation 3 for various problems is shown in Table 7.2. The problems are simulated with TML-3 with the default time step size of 5 ms, and 5 MGCMFD steps per transport step. The SDC method is used to solve the EPKEs. The 51-group library is used for the pin problem and the 4-mini test problems [68]. The 47-group library is used for the SPERT problem. The abbreviation **MRE** represents the relative error with the largest absolute value to the reference results, while **MAR** stands for the absolute error with the largest



(a) **(b)**

Figure 7.9: Comparison of approximations used for the CMFD coefficients calculation in TML for the 4-mini HFP test problem. (a) plots the reactivity profile on the CMFD level for different approximations. It can be seen that approximation 3 underestimates the reactivity when control rod is withdrawn out of the core. Several dips can be observed for approximation 3. They are at the start points of the CMFD level. (b) plots the power profile. Due to the reactivity underestimation, it can be seen that the power also gets underestimated. In contrast, the reactivity profile and the power profile of approximation 2 are much more accurate.

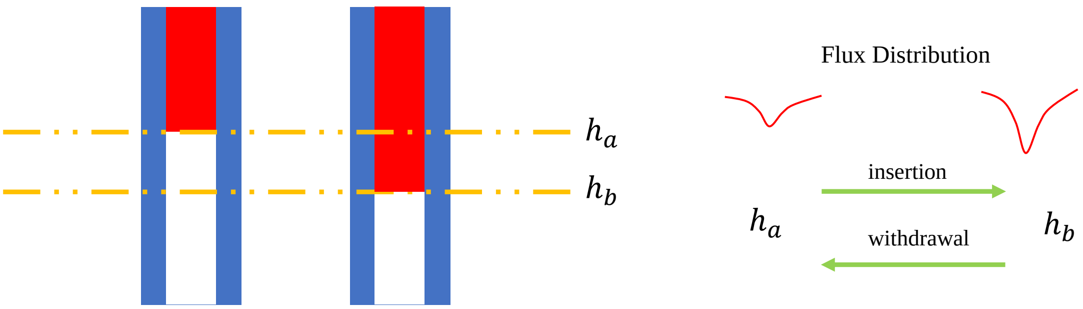


Figure 7.10: Illustration of deficiency of approximation 3 in [54]. When control rod tips move from h_a to h_b in the period [t_0,t_5], the flux at h_a keeps decreasing. Therefore, using the flux at t_5 to homogenize the cross section will underestimate the absorption. When control rod tips move from h_b to h_a in the period [t_0,t_5], the flux at h_b keeps increasing. Using the flux at t_5 to calculate the CMFD coefficients will overestimate the absorption.

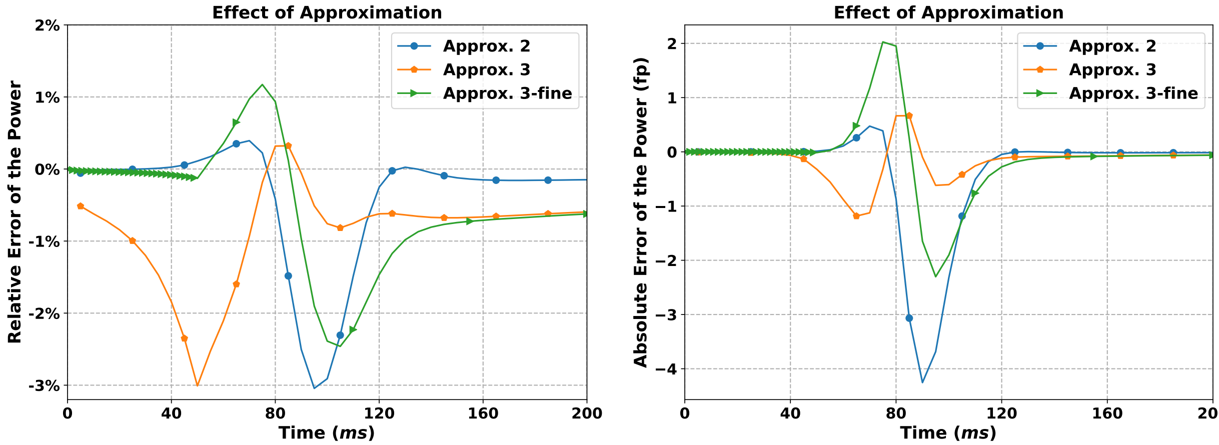
absolute value. The unit of absolute error is full power (fp). For the 4-mini test problems, approximation 2 performs much better. And in other cases, approximation 2 has similar performance to approximation 3. Approximation 2 and approximation 3 have comparable accuracy in terms of the relative error with the largest absolute value. In terms of the absolute error with the largest magnitude, approximation 3 does perform better, but this is due to the error cancellation.

Figures 7.11 investigate the errors of the power results. For approximation 3, as suggested by case "Approx. 3", the power gets underestimated during the rod movement ($t < 50$ ms), and the relative error reaches its maximum magnitude at $t = 50$ ms. After the rod movement, the magnitude of the relative error keeps decreasing till $t = 70$ ms, and is bounded by 1% after $t = 75$ ms. Using a time step of 0.2 ms, as suggested by the plot of "Approx. 3-fine", helps mitigate the issue of power underestimation. But, the magnitudes of both the absolute error and relative error for $t \geq 75$ ms get amplified, compared to the case with constant time step. The results indicate that the maximum magnitude of the absolute error for the case "Approx. 3" is smaller than that for the case "Approx. 3-fine", so it is concluded this is due to error cancellation. The magnitude of the relative error for the results of approximation 2 during the rod movement is much smaller than that of approximation 3. However, after the control rod movement, both the relative error and the absolute error of the results reach their largest magnitude near t=100 ms. The pattern of errors observed in case "Approx. 2" is similar to that observed in case "Approx. 3-fine". Therefore, it may be concluded that the major reason that approximation 2 outperforms approximation 3 for the SPERT cases is that the error from underestimating the power during the control rod movement offsets the error in the feedback stage.

Table 7.2: Accuracy comparison of approximation 2 to approximation for various problems. The description of the pin problem and the SPERT problems can be found in Appendix B.

Problems	MRE		MAR (fp)	
	Approx. 2	Approx. 3	Approx. 2	Approx. 3
Pin	-0.5%	-1.22%	-0.05	-0.06
SPERT-2D-HFP	-1.45%	-1.08%	-0.39	0.11
SPERT-3D-HZP	-3.04%	-3.00%	-4.26	-1.18
SPERT-3D-HFP	-3.02%	-2.89%	-2.78	-0.82
4-mini-HZP	-2.09%	-14.5%	-2E-3	0.02
4-mini-HFP	-2.46%	-17.04%	-0.13	-0.63

Moreover, using approximation 2 helps to reduce the computational cost from cross sections calculation. In MPACT, for problems where the control rods are heterogeneous axially and modeled explicitly, a time-consuming resonance self-shielding calculation [66] is performed when the rod position changes. When approximation 3 is used to calculate the CMFD coefficients at t_0, cross sections at t_0 must be either recalculated or pre-calculated and stored in memory, since the flux is calculated with the cross sections at t_5. MPACT chooses to recalculate the cross sections to reduce the memory burden at the cost of efficiency. On the contrary, when approximation 2 is used, the CMFD coefficients at t_0 can be calculated at the beginning of each step. Therefore, there is no need for recalculation. For the problems shown in Table 7.3, the run time reduction ranges



(a) **(b)**

Figure 7.11: Error evolution of approximation 2 and approximation 3 for the SPERT-3D-HZP case. (a) shows the evolution of the relative error, while (b) shows the evolution of the absolute error. The cases "Approx. 2" and "Approx. 3" use a constant time step of 5 ms, while the case "Approx. 3-fine" uses a much finer time step size $\Delta t = 0.2$ ms during the control rod movement. The plots indicate that the using finer time steps for approximation 3 can reduce the errors during the control rod movement, but the maximum magnitude of the absolute error gets increased. The comparison verifies that the error cancellation helps to make approximation 3 outperform approximation 2 in terms of the magnitude of the absolute value.

from 3% to 5%. However, it should be much more significant in larger problems that model the SCRAM.

Table 7.3: Run time comparison of approximation 2 to approximation 3. The run time are given in hours.

Method Problem	Approx. 2	Approx. 3
4-mini HZP	4.69	4.55
4-mini HFP	5.11	4.97
7× 7 Test Problem	46.07	44.15

Since we cannot always rely on the error cancellation from approximation 3, and approximation 3 is less efficient, approximation 2 should be set as the new default method to calculate the CMFD coefficients in MPACT.

7.2.2 New 1GCMFD Level Implementation

When approximation 2 is used, the MGCMFD flux at t_5 from the previous MOC step is close to the MGCMFD flux at t_0 for each transport step. Therefore, the temporal variation of MGCMFD flux and 1GCMFD system is smoother. Given this, it is possible to develop a new 1GCMFD level implementation in TML-4.

For the new implementation, it is assumed that the shape function of the MGCMFD flux varies linearly in space. Then the 1GCMFD coefficients on the 1GCMFD level are obtained from collapsing with the interpolated normalized MGCMFD flux, i.e.:

$$\Sigma_{x,m,g}(t_{n''}) = \sum_{g=1}^{G} \Sigma_{x,m,g}(t_{n''})\Phi^P_{m,g}(t_{n''}) \, , \; x = t, s, f, a \, , \tag{7.60}$$

with

$$\begin{gathered} Y_{m,g}(t_{n''}) = Y_{m,g}(t_{n'})\eta + Y_{m,g}(t_{n'+1})(1-\eta), \eta = \frac{t_{n''} - t_{n'+1}}{t_{n'+1} - t_{n'}} \, , \\ Y = \Phi^P, \tilde{\Sigma}_x \, , \end{gathered} \tag{7.61}$$

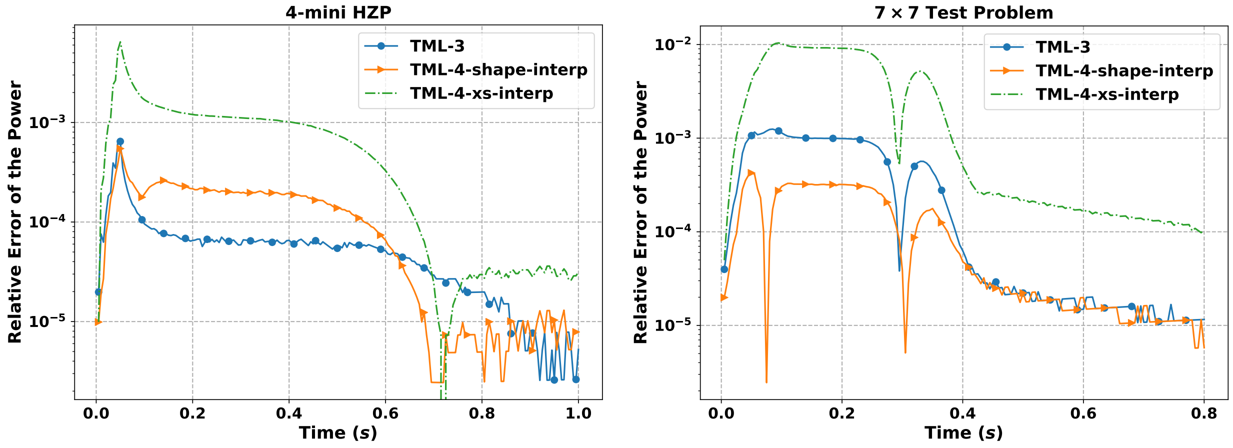
where

$$\Phi^P_{m,g}(t_{n'}) = \frac{\Phi_{m,g}(t_{n'})}{\sum_{g=1}^{G} \Phi_{m,g}(t_{n'})} \, . \tag{7.62}$$

Then after solving the 1GCMFD TFSP, the MGCMFD flux is updated by

$$\phi^C_{g,m} = \frac{\Phi^C_m}{\sum_{g=1}^{G} \phi^P_{g,m}} \phi^P_{g,m} \, . \tag{6.9 Revisited}$$

Figures 7.12 compare the TML-4 with the new implementation (shape-interp) to the TML-4 in Chapter 6 (xs-interp) and the TML-3 with default level settings in the 4-mini HZP problem and 7×7 test problem. The reference solution is obtained using TML-3 with 10 CMFD steps per transport step. Approximation 2 is used and the time step size is 5 ms. It is observed that TML-4 performs about as well as the TML-3 in the 4-mini HZP problem, but produces more accurate results than TML-3 in the 7×7 problem. Therefore, with approximation 2 and the new implementation of the 1GCMFD level, there is no need to use the TML-4 in a hybrid way as suggested in Chapter 6.



(a)

(b)

Figure 7.12: Investigation of TML-4 with the new implementation of 1GCMFD level. The new TML-4 results are compared to the old TML-4 and the default TML-3 results. (a) shows the comparison results in the 4-mini HZP test problem, and (b) shows the comparison results in the 7×7 test problem. For the new implementation, the shape of the MGCMFD flux is interpolated in time. The results are indicated by "TML-4-shape-interp". The results of the old implementation of TML-4 are indicated by the "TML-4-xs-interp". It can be seen that the new implementation is much more accurate than the old implementation, and can be more accurate than the default TML-3.

7.3 Operator Splitting Scheme

In MPACT, the feedback from TH and neutronics are not fully coupled during the simulation. The OS technique is used to perform the transient simulations with feedback from other physics. The details of this were briefly covered in Section 7.2.1. MPACT solves the NTE first to provide the heat distribution for the TH solver, and the TH solver, in turn, solves the TH field and provides updated cross sections to the MPACT. The NTE and the TH are both solved holding the variables from other fields constant first and then coupled together. This operator splitting technique is an example of the LS method and is, in general, first-order accurate in terms of the time step size. The flowchart of the coupling scheme is shown in Figure 7.13.

For the coupling scheme based on Lie Splitting (LS), it has been expected that increasing the time step size will increase the error and decreasing the time step size will make the solution more accurate. This expectation is valid when the problems are simulated with *constant time steps*. However, it is observed that during the simulation, if the time step is varied by either increasing or decreasing it, the solution is made less accurate.

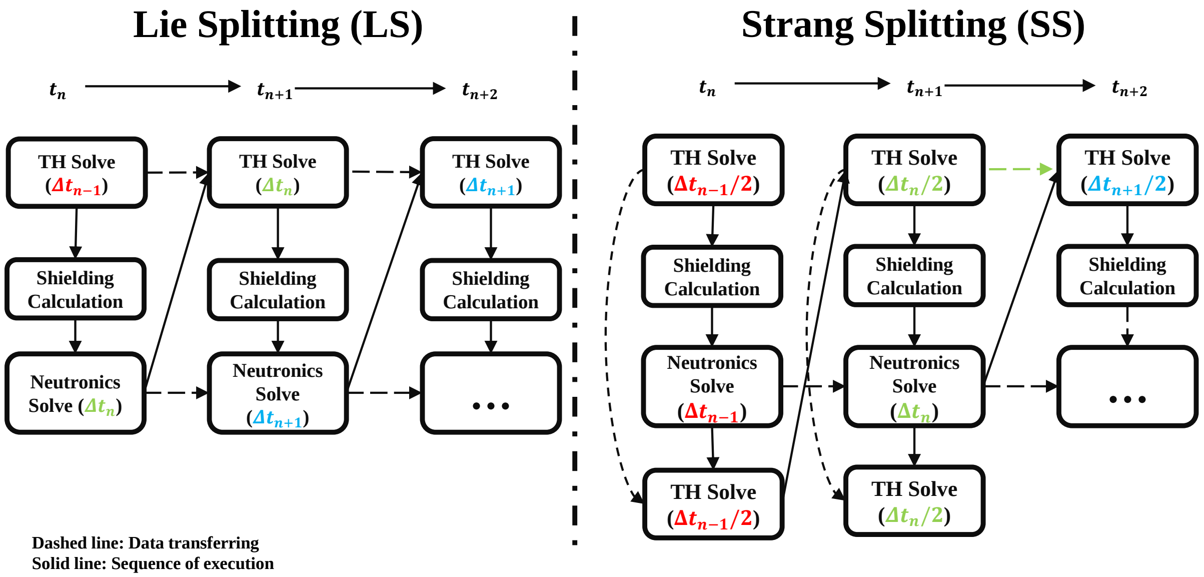

Lie Splitting (LS)
Strang Splitting (SS)
t_n
t_{n+1}
t_{n+2}
TH Solve (Δt_{n-1})
TH Solve (Δt_n)
TH Solve (Δt_{n+1})
Shielding Calculation
Neutronics Solve (Δt_n)
Neutronics Solve (Δt_{n+1})
...
TH Solve ($\Delta t_{n-1}/2$)
TH Solve ($\Delta t_n/2$)
TH Solve ($\Delta t_{n+1}/2$)
Neutronics Solve (Δt_{n-1})
Neutronics Solve (Δt_n)
Dashed line: Data transferring
Solid line: Sequence of execution

Figure 7.13: Multiphysics coupling scheme for transient simulation used in MPACT. The flowchart on the left side is the illustration of the original coupling scheme of MPACT and TH. The flowchart on the right side is the new coupling scheme based on SS.

7.3.1 Evidence from PKE-EF Problem

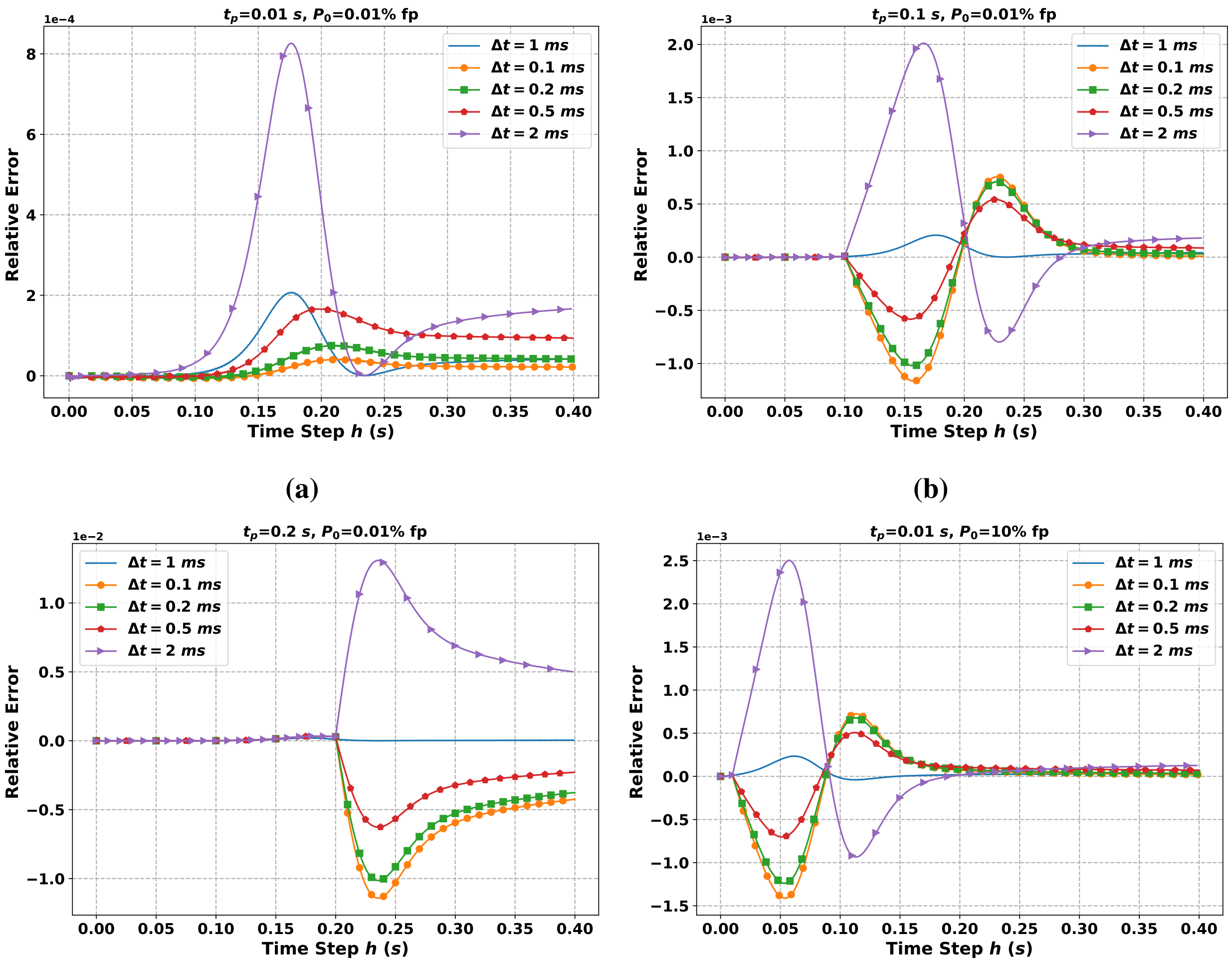
We first notice the lack of consistency with time step size for the LS in the PKE-EF problem, where the LS is used to couple the PKE to the energy feedback. In the PKE-EF problem, the PKEs are solved with the SDC method, and the energy deposition is calculated by

$$Q^{(n)} = Q^{(n-1)} exp(-\lambda_H \Delta t_{n-1}) + \frac{1 - e^{-\lambda_H \Delta t_{n-1}}}{\lambda_H} P^{(n)} = A(\Delta t_{n-1}) Q^{(n-1)} + B(\Delta t_{n-1}) P^{(n)} \,. \tag{7.63}$$

n is used to index the time step here. Therefore, both the PKE model and energy deposition problem are accurately solved, and the error is from operator splitting.

Figures 7.14 show the effect of varying time steps when LS is used. The solution of PKE-EF with the SDC-IMEX solver is used as the reference. It can be observed that either coarsening or refining the time steps during the calculations amplifies the error significantly.

Decreasing the time step size generally makes the error more negative while increasing the time step size positive. The only exception is when the time step size is perturbed at 0.01 ms for the problem with $P_0 = 0.01\%$ fp. This error only gets amplified when the time steps are coarsened. However, if P_0 is modified to be 10% fp, the inconsistency between the error and the time step size can be observed again.



(a)

(b)

(c)

(d)

Figure 7.14: Inconsistency between error of power and time step size in Lie Splitting in PKE-EF problems. The initial reactivity is \$1.2. The default time step size is 1 ms. The time step changes to 0.1 ms, 0.2 ms, 0.5 ms and 2 ms at t_p. t_p in (a)–(d) is 0.01 s, 0.1 s, 0.2 s, 0.01 s. The initial value P_0 in (d) is 10% fp, while in other three sub-figures are 0.01% fp. In general, varying the time step size will amplify the absolute value significantly.

The reason is that when a constant time step is used, the LS technique is similar to the Strang Splitting (SS), a second-order OS technique that has almost the same computational complexity. The flowchart of the SS implementation is also shown in Figure 7.13. In the scheme, for each time step, the TH is solved first with a half time step and provides the TH field to the neutronics solver. The neutronics solver advances the neutron solution with a full time step, and the governing equation for the TH problem is solved again with a half time step. When SS is used, the very first half step of the solution of the TH can be omitted, as a perturbation has not been introduced. Consequently, the power distribution and TH are constant over the time step. After the first half step, the two TH solutions can be combined, if error in solving the TH problem is small. For the

PKE-EF model, it is very simple to see that:

$$A(\Delta t)Q^{(n-1)} + B(\Delta t)P^{(n)} = \\ A\left(\frac{\Delta t}{2}\right)\left(A\left(\frac{\Delta t}{2}\right)Q^{(n-1)} + B\left(\frac{\Delta t}{2}\right)P^{(n)}\right) + B\left(\frac{\Delta t}{2}\right)P^{(n)}\,, \tag{7.64}$$

i.e. the results after two half-steps are the same as those after one time step. Thus the SS scheme is simplified to the LS.

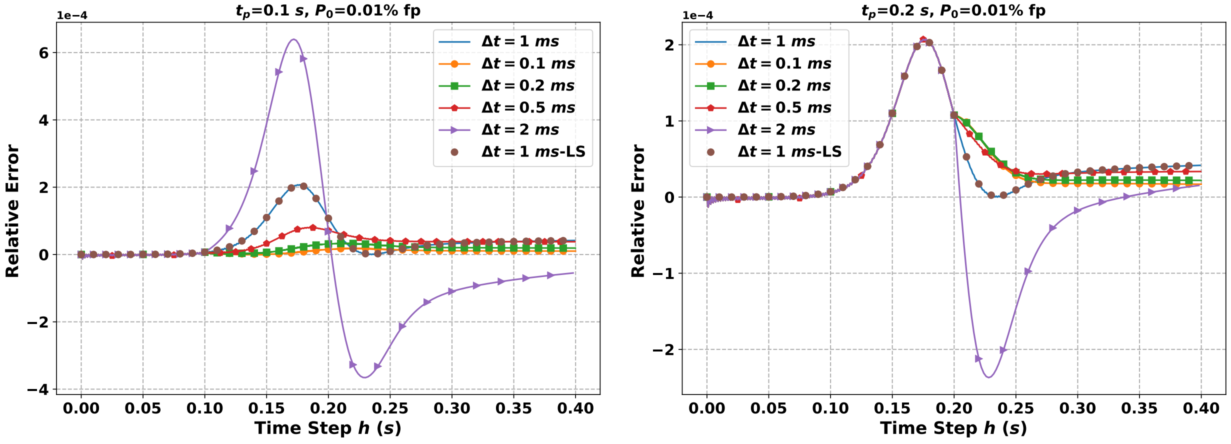
When the time step changes, the LS is no longer similar to SS. For the n^{th} time step, the step size to solve the TH equations is Δt_n when using the LS technique, but should be $(\Delta t_n + \Delta t_{n+1})/2$ to enforce equivalence of LS to SS. Therefore, a temporal error $(\Delta t_n - \Delta t_{n+1})$ appears for the LS method and the coupling error is no longer second-order accurate. The method becomes be first-order accurate, and the observed error gets amplified instantly.

To have more consistent behavior between the error and time step size, the SS is used, and the application is verified with the PKE-EF model. The results are shown in Figures 7.15. It can be observed that when constant time steps of 1 ms are used, SS and LS can produce the results with the same error. However, when the time step size changes before the power peak around 0.15 s, reducing the time step size reduces the absolute value of the error and increasing the time step size amplifies the magnitude of the error. When the time step size is adjusted after the power peak, reducing the time step size makes the error accumulate more slowly while increasing the time step size amplifies the magnitude of the error. There is no sudden amplification of the error after the time step size modification.

7.3.2 Evidence from MPACT Simulations

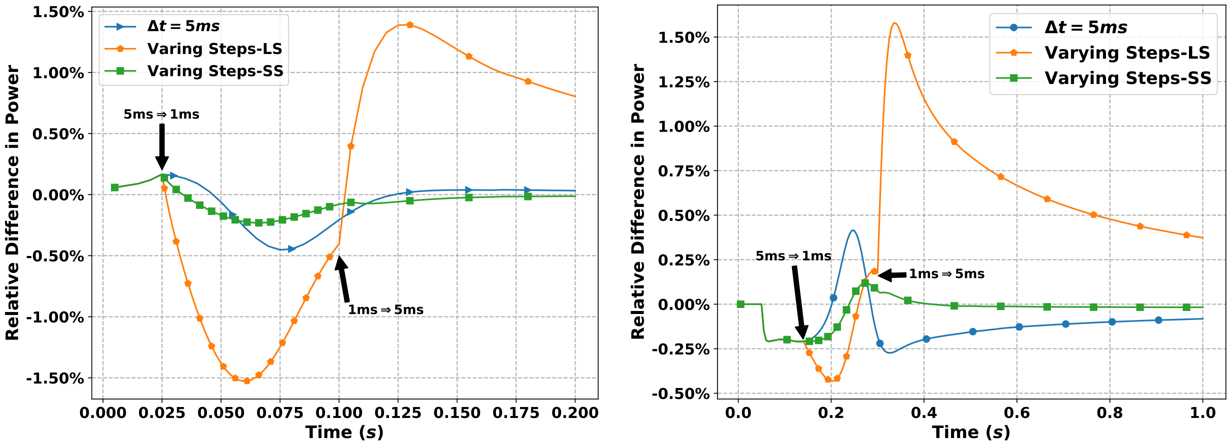
Practical examples from a 2D pin-cell transient problem and the 4-mini regression test problem are used to show the deficiency of LS. A maximum reactivity of \$1.0278 is inserted for the pin problem. The results are shown in Figures 7.16. It can again be seen that either decreasing the time step size or increasing the time step size will make the solutions less accurate, and the relative error can be amplified by orders of magnitude. When SS is used, the error change is smooth. Reducing the step size before the power peak reduces the peak error.

Figures 7.17 show the relative difference between the results of using LS and SS when a constant time step is used. It can be observed that the magnitude of the relative difference is quite small– less than 0.1% for the cases investigated here. The practical results using LS and SS are not the same because the TH solver is first-order accurate, and the pin power distribution is calculated with the cross sections that do not get updated during the TH calculations.



(a) **(b)**

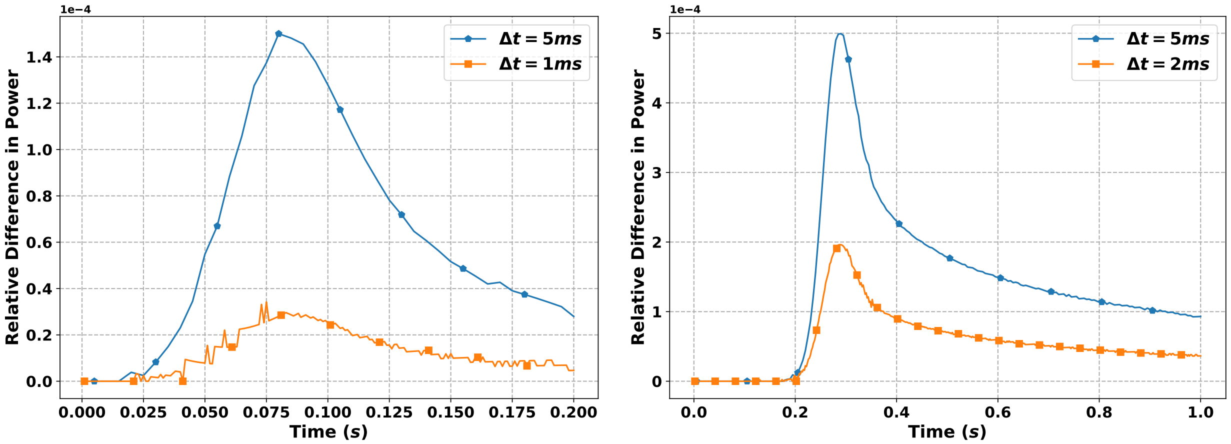
Figure 7.15: Error for varying time steps in SS for PKE-EF problem. The initial reactivity is $1. The default time step size is 1 ms. The time step changes to 0.1 ms, 0.2 ms, 0.5 ms and 2 ms at t_p. t_p is 0.1 s and 0.2 s in (a) and (b), respectively. The initial value P_0 is 0.01% fp. It can be seen that with finer time steps, the error will not get amplified. When the time step size perturbation is applied before the power peak near 0.15 s, using a finer time step size will make the error smaller.



(a) **(b)**

Figure 7.16: Comparison of LS and SS with varying time step sizes. (a) plots the results for a pin transient problem and (b) shows the results of the 4-mini HZP test problem. When Lie Splitting is used, altering time steps amplifies the error amplified, while for Strang Splitting , the error changes smoothly.

To summarize, the error from OS is inconsistent with the time step size when the LS is used and the time step size is modified. Therefore, LS is not suitable for a scheme where the time step sizes are adjusted for some time periods, and especially for an adaptive-stepping scheme. As a result, *the LS should never be used, and the coupling scheme should be modified to use Strang*



(a)

(b)

Figure 7.17: Comparison of LS and SS with constant time step sizes. (a) plots the results for a pin transient problem and (b) shows the results of the 4-mini HZP problem. When constant time steps are used, the relative difference between the LS and SS results are quite small compared to the relative error of the power results observed in Figure 7.16.

Splitting shown in Figure 7.13. For the simulations presented later in this thesis, Strang Splitting is used.

7.4 Summary

In this chapter, we used the SDC method, a stable, robust, and efficient high-order time-integration method for the initial value problem to solve the EPKE problems in TML. A possible new approach for calculating the residual function was proposed and implementation of the low-order solvers in SDC for PKEs was presented. It is observed that SDC can achieve high-order accuracy without loss of stability.

Next, the TML scheme was decoupled from the TH feedback. This modifies the approximation on the CMFD coefficients at the left endpoint of the CMFD level. The new approximation reduces the error of the simulation when the control rod is moving. A new implementation of the TML-4 scheme was then proposed, and the new scheme is much more accurate than that proposed in Chapter 6.

Finally, we found that the current multiphysics scheme in MPACT based on Lie Splitting is deficient via examples from a simple PKE with linear energy feedback model, and the numerical simulation results with MPACT. Strang Splitting was introduced as an alternative OS for multiphysics simulations, and demonstrated to be superior to LS.

It should be noted that using SS to couple the neutronics and feedback from TH may be just a temporary approach. As suggested by [30], the ultimate goal for the multiphysics simulations should be solving the fully coupled neutronics-TH problem implicitly with a high-order solver. As shown in Section 7.1, the SDC method is a promising approach, and further research on this topic is recommended.

CHAPTER 8

Adaptive Time-stepping Methodology for Transient Multilevel

This chapter introduces the adaptive time-stepping (ATS) methodology developed to further improve the efficiency of transient calculation with the Transient Multilevel (TML) scheme. Like most adaptive methods, it is constructed in such a way that the error is predicted and then kept below a desired threshold.

In the development of the ATS methods, two errors are controlled. One is from the solution of neutronics and is referred to as the neutronics internal error. The other is from the coupling via operator splitting (OS), and is referred to as the coupling error. These errors are predicted by the reduced-order models, that are developed based on point-kinetics equations (PKEs). Then the time step size is calculated, so the predicted errors are smaller than the error tolerances. Since there is some discrepancy between the error tolerances and the maximum errors observed for practical simulations, a parametric study on the error tolerances is needed. The goal in this chapter is to let the maximum relative error observed for practical simulations smaller than 1%.

The remainder of the chapter is organized as follows. To start, in Section 8.1 an overview of the existing ATS methods is provided. The limitations of these methods are discussed. Therefore, we use the PKE models to estimate the errors. Section 8.2 then describes the development of the ATS method for controlling the neutronics internal error, while Section 8.3 introduces the development of the ATS approach for controlling the coupling error. In both sections, the ATS methods are tested in small cases. The overall flowchart of the ATS methods is shown in Section 8.4. Since the time step size keeps varying, an adaptive CMFD level in TML is necessary. Therefore, an adaptive method to adjust the number of CMFD steps per transport step is also developed and presented. The methods developed in this chapter are then applied together in more realistic problems and the results are presented in Section 8.5. Finally, in Section 8.6, a summary is provided.

8.1 Existing ATS Methods

Compared to efforts developing the methodology for solving the time-dependent NTE, significantly less effort has been made to develop the ATS methods. In the works that have been published, most of the research is based on the calculation of the derivatives [118, 119, 120]. The basic idea of these methods is to obtain a relation between the local truncation error and the time step size. Then the time step size is determined with a given error tolerance.

To illustrate this process, let's assume the initial value problem is written as

$$\mathbf{y}' = \mathbf{F}(\mathbf{y}, t)\,, \tag{8.1}$$

and is solved with a q^{th}-order accurate method in time. If no error is from spatial discretization, the error is then only from time discretization, and expressed as:

$$\tau = O(h^{q+1}) = \mathbf{C}\mathbf{y}^{(q+1)} h^{q+1}\,, \tag{8.2}$$

where τ is the local truncation error. $\mathbf{y}^{(q+1)}$ is the $(q+1)^{th}$ derivative of $\mathbf{y}$. h is the time step size. In practical simulations, $\mathbf{y}$ is a vector and $\mathbf{C}$ is a diagonal matrix representing the undetermined coefficients.

Given a tolerance on the local truncation error, ϵ, the time step size is calculated by

$$h = \left(\frac{\epsilon}{||\mathbf{C}\mathbf{y}^{(\mathbf{q+1})}||_p}\right)^{1/(q+1)}\,, \tag{8.3}$$

with the L_p norm $||\mathbf{C}\mathbf{y}^{(\mathbf{q+1})}||_p$. An alternative way for calculating the time step size is to calculate the time step size growth ratio by:

$$\frac{h_{n+1}}{h_n} = \left(\frac{\epsilon}{||\mathbf{C}\mathbf{y}^{(\mathbf{q+1})} h_n^{q+1}||_p}\right)^{1/(q+1)}\,, \tag{8.4}$$

where n is the index for the time step. More complex models have also been developed in [121].

Whatever the procedure is, the derivative is needed. The major deficiency of this approach is that only the reduced-order information, that is the norm of the predicted error, is used. Therefore, these existing ATS methods are applied in some semi-quantitative way though are developed quantitatively. And, there are limitations for the application of these procedures to TML.

First, approximated derivatives may have significant errors for transient problems, especially if the solutions change very fast. What has been observed is that the predicted time step sizes are always oscillating [120]. Using the alpha eigenvalue can help to stabilize the oscillation of the predicted time step sizes [119] for some simple problems. However, in more realistic problems, the oscillation is still observed, even when the transport operator varies smoothly (i.e., when the control rods stop moving) [120].

Moreover, the direct application of Eq. (8.2) is often inconsistent with the time-integration method analyzed, thus unnecessary. In real simulations, approaches such as shape transformation [122, 123, 124], or source splitting techniques [125, 126, 43] are introduced to stabilize the scheme and reduce the error. These approaches fundamentally change the equations being solved. For example, it has been observed that using the shape transformation without TML can reduce the maximum relative error in magnitude by more than a factor of 100 [51] compared to the scheme without shape transformation. The only reason is that the $||\mathbf{Cy^{(q+1)}}||_p$ has been reduced significantly.

For complex multilevel methods like TML, the task of defining and calculating $\mathbf{C}$ is even more challenging. For example, researchers in [127] have developed some complex expressions to estimate Θ for the IQS method. Θ is defined by:

$$\tau = \Theta h^{q+1} \,. \tag{8.5}$$

They have shown that the errors τ estimated by different methods for Θ are different in orders of magnitudes [127].

Last but not least, the basic assumption for these methods is that the time-dependent problem is solved with the neutronics solver fully coupled to the TH solver. However, OS has been introduced to couple the TH solver and the neutronics solver. Therefore additional error from OS has been introduced. The error from OS depends on the commutator of the operators, and it vanishes when the operators involved are commutative [57]. For real problems, this is never the case. Therefore the attempts to define C or Θ may be made on a "false" problem without considering the errors from OS [120].

Compared to these "quantitative" methods, using PKE models to estimate the error is a straightforward approach, despite the shortcomings of using simpler models. These models are used to estimate the error from solving the NTE and the error from OS, and are introduced in the next two sections.

8.2 ATS Method for Neutronics Solver

In this section, we will introduce how the neutronics internal error is calculated and how the time step size is determined for a given tolerance. Then simple numerical results are provided to verify the method.

8.2.1 Neutronics Internal Error Estimation

The procedure to estimate the neutronics internal error is presented in this part. The relative error in the power growth ratio is introduced to characterize the local error. We do not use the term "local truncation error" because the error investigated is not obtained analytically.

The power growth ratio is calculated by

$$r_h(t_n) = \frac{P(t_n)}{P(t_n - h)} , \tag{8.6}$$

where h is the time step size.

The relative error of power ratio can be obtained directly from MPACT simulations. It can also be obtained by a PKE model. Numerical results shown in Figures 8.1 will demonstrate that the errors estimated from MPACT and the PKE model are in good agreement. Before this, how the relative errors in the power growth ratio are obtained for these two methods are introduced.

8.2.1.1 Error in Power Growth Ratio

Two problems, which are a pin transient problem and the 4-mini test problem, are used. For MPACT results, the reference case is simulated with time steps of 0.2 ms for the pin transient problem, and 0.5 ms for the 4-mini test problem. The time step sizes of the reference cases in different problems are different to save time. For the same reason, the 8-group library is used in the 4-mini test problem, while the 51-group library is used for the pin problem. The test cases are simulated with coarser time steps. Then the local error at t_n for the test case simulated with time step size h is calculated by

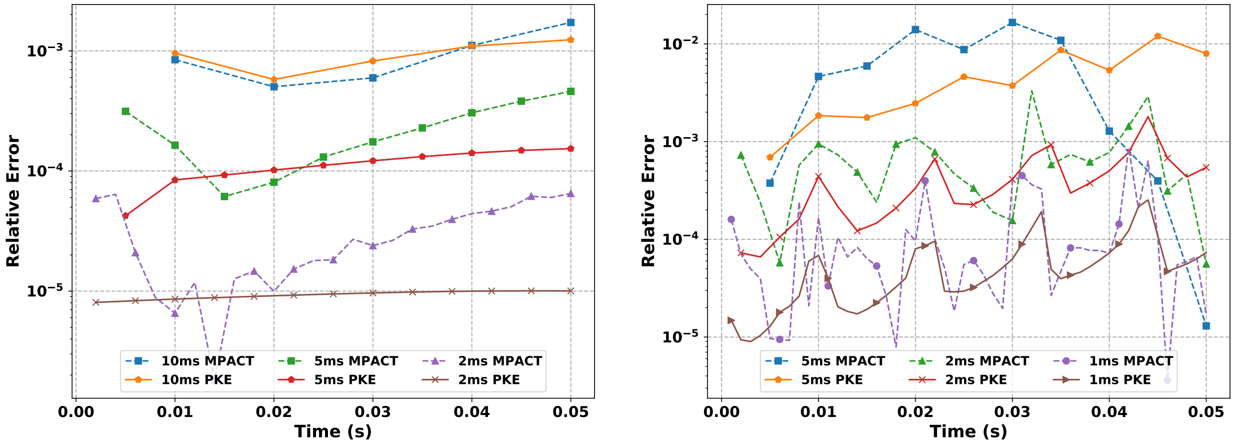
$$e_{mpact}(t_n) = |\frac{r_h(t_n)}{r_{ref}(t_n)} - 1| . \tag{8.7}$$

For the PKE results, the relative differences in the solutions of two PKE problems are obtained. One PKE problem uses a polynomial reactivity profile. The polynomial is obtained by using a quadratic or cubic polynomial to fit the discrete reactivity values computed in the CMFD level of TML. The other problem uses a linear reactivity profile, assuming the reactivity varies linearly within the time step. The other PKE parameters for each time step are the latest EPKE parameters in the transport level in the TML scheme. The solution of the PKE system with polynomial reactivity is denoted as p_f, and the solution of the PKE system with linear reactivity is denoted as p_l. Then the relative difference is calculated by:

$$e_{pke}(t_n) = \left|\frac{p_l(t_n)}{p_f(t_n)} - 1\right| . \tag{8.8}$$

The PKEs are solved with the SDC solver.

The e_{pke} (labeled as "PKE") and e_{mpact} (labeled as "MPACT") for each case are plotted and shown in Figures 8.1. Several observations can be made:



(a)

(b)

Figure 8.1: Investigation of characterization of the neutronics internal error. The absolute values of e_{mpact} and e_{pke} are shown. (a) shows the results for the pin problem. (b) shows the results of the 4-mini test problem.

- For each test case, the evolutions of e_{pke} and e_{mpact} are similar. For the time steps where e_{mpact} is a local maximum, e_{pke} is also a local maximum or close to the local maximum. This indicates the model is suitable. There is no axial heterogeneity for the 2D pin cases. Therefore, the error evolution in the pin problem is smoother. The 4-mini test problem is a 3D problem with control rod movement modeled explicitly. The oscillation of the error is mainly due to the discontinuity in moving materials and the rod decusping.

- For each test case, e_{pke} and e_{mpact} are similar in orders of magnitudes. But the errors estimated by the PKE model are smaller than the errors observed in MPACT for most time steps. The reason is that the error estimated here is more like the error in the amplitude function, and the estimation ignores the error for the shape function.

- The error of the power growth ratio for each step can vary by orders of magnitude in a short period. This is due to the axial heterogeneity.

From this assessment, the relative difference predicted by the PKE model is considered to be a reasonable characterization of the local error at each step.

8.2.1.2 Final Error Estimation

In this part, the estimation of the errors to be controlled for the ATS is presented. Two errors needed to be controlled. One is the one-step error, that is denoted by e_1^{in}. The other is the multi-step error, that is denoted by e_m^{in}. The superscript ^{in} stands for "internal".

The one-step error is the relative difference in the power ratio predicted by the PKE model presented in Section 8.2.1. The multi-step error is obtained for the maximum relative difference in magnitudes of the solutions of the two PKE systems for a reasonably long time. The PKEs are solved with time steps of Δt_n. Therefore, multiple steps used performed. The reactivity profile inside each time step keeps unchanged. In this chapter, the time range in which the multi-step error is predicted is referred to as the "prediction horizon".

Controlling the one-step error is similar to controlling the local truncation error. However, simply controlling one-step error is not enough, because the error is accumulated. For example, if the tolerance for the one-step error is 0.001 and the estimated time step size is 1 ms, then in 10 ms, the accumulated relatively error could be $(1 + 0.001)^{10} - 1 \approx 0.01$, indicating that an additional 1% uncertainty is accumulated in the simulated results for every 0.01 s of simulation.

Algorithm 15 Estimation of the errors to be controlled in TML.

```
Input:time step size $\Delta t_n$.
Output:one-step error $e_1^{in}$ and multi-step error $e_m^{in}$
For the $n^{th}$ time step, fit the reactivity with high-degree polynomial $f(t)$ and obtain the linear reactivity profile $f_l(t)$.
Determine the number of time steps by
    $$N = \min\left(\max\left(10, \lceil \frac{0.05}{\Delta t_n} \rceil\right), 20\right). \quad \text{(Alg 15.1)}$$
Solve the PKE with the reactivity profile $f(t)$ for $N$ steps, with $t \in [0, \Delta t_n]$ for each step. Store the power profile $p_f$.
Solve the PKE with the reactivity profile $f_l(t)$ for $N$ steps, with $t \in [0, \Delta t_n]$ for each step. Store the power profile $p_l$.
Calculate the relative difference of results in the first step, $e_1^{in}$, and the maximum relative difference in magnitude, $e_m^{in}$, in the prediction range.
```

The algorithms to estimate the errors are presented in Algorithm 15. The prediction horizon is 50 ms. However, bounds are imposed on the number of prediction steps performed. The lower bound 10 is selected so that when the time step size is large, we still perform some estimation of the multi-step error. A relatively small time step size indicates the error is relatively large from the previous step, i.e., the perturbations to the system are very fast. In this case, the perturbation will be significant after 50 ms, so there is no need to perform the estimate for this long time. 20 is used as the upper bound for the number of prediction steps for the multi-step error.

When N steps are performed for the prediction, the ratio e_m^{in}/e_1^{in} can be far smaller than N. Figure 8.2 illustrates the ratio e_m^{in}/e_1^{in} when different time step sizes and the number of prediction steps are used to estimate the errors in the transient pin problem. The maximum reactivity is reached after 0.05 s and is \$1.43. It can be seen the ratio is not equal to the steps performed initially. After 0.04 s, the ratio approaches the number of steps. An additional case with $\Delta t = 1$ ms and $N = 50$ is performed. The maximum reactivity for the new case is \$0.53 and is reached after 0.05 s. It is seen that the ratio is well below 50.

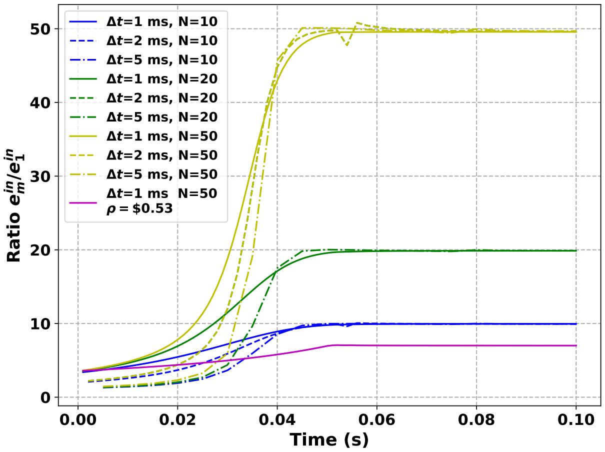


Figure 8.2: Ratio of the multi-step error to the one-step error for a pin problem. The first nine cases are from a pin problem with a maximum reactivity of \$1.43. The reactivity of the last case is reduced to \$0.53. The maximum reactivity is reached at 0.05 s. When the reactivity is small, the ratio between e_m^{in}/e_1^{in} is smaller than N.

8.2.2 Step Size Calculation

This part presents the method to calculate the time step size.

8.2.2.1 Determining Time Step Size from One-step Error

For the one-step error, if the PKE system with polynomial reactivity is considered as the reference and the PKE system with linear reactivity is considered as an approximation, then the local truncation error can be shown to be $O(\Delta t^3)$. Given a tolerance ϵ_1^{in}, for the one-step error that is estimated with the time step size Δt_n, the time step size can be estimated by:

$$\Delta t_{n+1}^1 = \left(\frac{\epsilon_1^{in}}{e_1^{in}(\Delta t_n)} \right)^{1/3} \Delta t_n \, . \tag{8.9}$$

Due to the heterogeneity of the problem, it is possible that the one-step error can vary too much. Therefore, it is necessary to evaluate the quality of the prediction. The quality of the prediction is determined by

$$\eta = \frac{\frac{e_1^{in}(t_{n+1})}{e_1^{in}(t_n)}}{\left(\frac{\Delta t_{n+1}}{\Delta t_n}\right)^3} \,. \tag{8.10}$$

$\eta = 1$ denotes that the prediction is relatively accurate, while $\eta \neq 1$ indicates that there are some effects the model cannot capture. Only $\eta > 1$ should be considered, because this indicates the prediction is too optimistic. To correct the time-step for an optimistic prediction, we use

$$\Delta t_{n+1}^{(1)} = \frac{\Delta t_{n+1}^{(1)}}{\max(\eta^{1/3}, 1)} \,, \tag{8.11}$$

for the correction of the predicted time step size. An analogous understanding of Eq. (8.11) would be the use of a "safety factor". This has been used to correct the time step size in many areas [128].

8.2.2.2 Determining Time Step Size from Multi-step Error

The error accumulated in the prediction horizon is $O(\Delta t^2)$. Given a tolerance ϵ_m^{in} for the multi-step error, the time step size is

$$\Delta t_{n+1}^{(2)} = \left(\frac{\epsilon_m^{in}}{e_m^{in}(\Delta t_n)}\right)^{1/2} \Delta t_n \,. \tag{8.12}$$

The assumption used is that if the local truncation error is third order, the global error is second order.

8.2.2.3 Step Size Variation Limit

Considering the next time step size determined by controlling both errors, we have

$$\Delta t_{n+1}^{(3)} = \max(\Delta t_{n+1}^{(1)}, \Delta t_{n+1}^{(2)}) \,. \tag{8.13}$$

However, for practical reasons, the step size should also be limited by other constraints.

First, the time step size cannot increase or decrease too fast. This practice is similar to what has been adopted in [45, 120, 128], and is a very common approach to control the time step size. For controlling the internal error,

$$\Delta t_{n+1}^{(4)} = \max\left(0.5\Delta t_n, \min(\Delta t_{n+1}^{(3)}, 1.5\Delta t_n)\right) \,. \tag{8.14}$$

The second limitation is imposed for the power growth ratio. The power growth ratio cannot be too large or too small. The reason is that the error in the calculation of reactivity is neglected for the PKE model presented here. And, the reactivity is obtained with the transport solution. When the power growth ratio is too large or too small, significant errors can be introduced in the estimation of the reactivity. To control this error, the power growth ratio is also limited by 10 and 1/10. This is just similar to the approach adopted in [45] to limit the $\omega\Delta t$ assuming the power can be expressed as $P_n exp(\omega\Delta t)$. Adopting this limit gives:

$$\Delta t_{n+1}^{(5)} = \min\left(\Delta t_{n+1}^{(4)}, \left|\frac{\Delta t_n}{\log_{10}\frac{P_l^1}{P_n}}\right|\right) , \tag{8.15}$$

where P_l^1 is the power solution for the first step in the prediction horizon. For most of practical simulations, the power growth ratio per step rarely violates the limit. If it does, it would indicate that the power will increase or decrease by a factor 10^5 in just 5 steps.

The third limitation is the minimum value for the time step size. The ultra-fine time step size can introduce instabilities and a negative source when solving the TFSP problem [129] for practical simulations. Therefore, the time step size should not be too small. In this thesis, the minimum time step size is set to be 1 ms. For the initialization, the first time step size is also set to be 1 ms. This avoids sophisticated calculations for the first time step. Furthermore, the perturbations for the first step always introduce singularities, so the smallest time step size is used to ensure minimal error. As a result, the eventual time step size is given by

$$\Delta t_{n+1}^{in} = \max(1\ ms, \Delta t_{n+1}^{(5)}) . \tag{8.16}$$

Here, Δt_{n+1}^{in} is the time step size predicted to control the internal error. The algorithm to determine the time step size is summarized in Algorithm 16.

8.2.3 Simple Numerical Results

The ATS method to control the neutronics internal error is now tested in the pin cell problem and the 8-group 4-mini test problem. The reference case is simulated with time steps of 0.2 ms for the pin transient problem, and 0.5 ms for the 4-mini test problem.

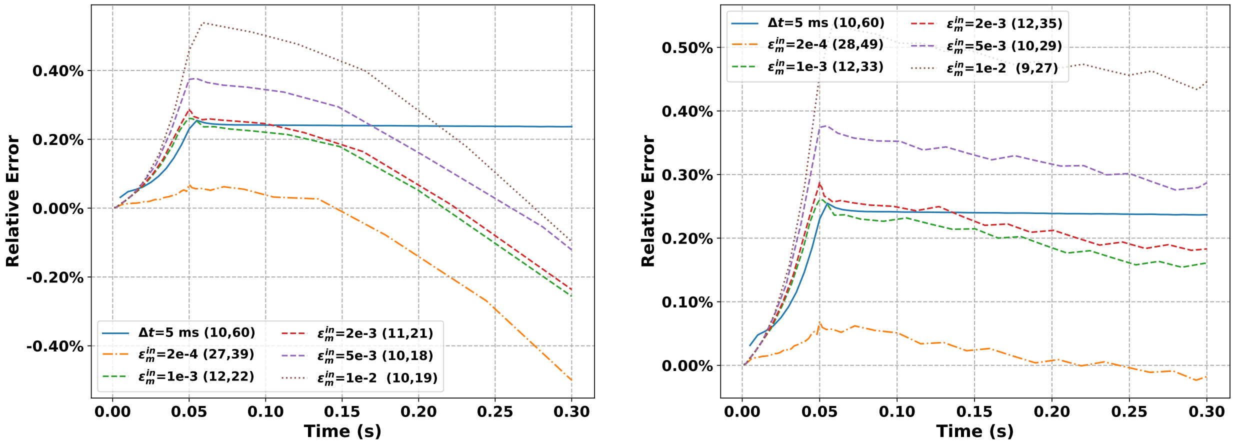
Algorithm 16 Time step size for control the neutronics internal error.

```
Input:time step size $\Delta t_n$, error tolerances $\epsilon_1^{in}$ and $\epsilon_m^{in}$.
Output:time step size $\Delta t_{n+1}^{in}$
1: Estimate the one-step error and multi-step error by Algorithm 15 for the time step $t_n$.
2: Calculate time step size $\Delta t_{n+1}^{(1)}$ with the provided error tolerance $\epsilon_1^{in}$ by Eq. (8.9). The safety
   factor is calculated to correct the time step size by Eq. (8.11).
3: Calculate time step size $\Delta t_{n+1}^{(2)}$ with the provided error tolerance $\epsilon_m^{in}$ by Eq. (8.12).
4: The time step size $\Delta t_{n+1}^{(3)}$ is obtained from $\Delta t_{n+1}^{(1)}$ and $\Delta t_{n+1}^{(2)}$ by Eq. (8.13).
5: The time step size variation limits suggested by Eqs. (8.14) to (8.16) are applied to $\Delta t_{n+1}^{(3)}$, and
   the time step size for the $(n+1)^{th}$ step is
```

$$\Delta t_{n+1}^{in} = \max\left(1\ ms, \min\left(\left|\frac{\Delta t_n}{\log_{10}\frac{P_l^1}{P_n}}\right|, \max\left(0.5\Delta t_n, \min\left(\Delta t_{n+1}^{(3)}, 1.5\Delta t_n\right)\right)\right)\right). \tag{Alg 16.1}$$

8.2.3.1 Transient Pin Problem

The results for the pin cases are shown in Figures 8.3. The problem is simulated for 0.3 s. Compared to the cases run for Figure 8.3a, the cases for Figure 8.3b limit the power growth ratio for each step by Eq. (8.15). For this problem, we use different ϵ_m^{in} and let $\epsilon_1^{in} = 0.2\epsilon_m^{in}$.



(a) **(b)**

Figure 8.3: Performance of the adaptive methodology for controlling the internal error of TML. The power growth ratio is not bounded in (a), while it is controlled in (b). When the power ratio is not bounded, the power can increase by a factor of 10^4 for every step. This inevitably induces the error in estimation of reactivity. Therefore the relative error keeps changing after $t = 0.05$ s in (a). The values in parentheses of the legend are the number of steps performed during the rod movement and the total number of steps for the whole transient process.

It can be seen that controlling the predicted errors does help to control the overall error when the reactivity is varying. Using a tighter tolerance leads to a smaller relative error in magnitude for $t < 0.05$ s. The maximum relative error observed at 0.05 s is larger than ϵ_m^{in}. This is because the estimated error is smaller than the actual error as suggested by Figure 8.1. This issue can be easily addressed by using a tighter tolerance.

For $t > 0.05$ s, it can be seen that the relative error in Figure 8.3a starts decreasing due to error cancellation. For this period, the PKE model cannot estimate the error accurately because the error is mainly from the reactivity calculated on the CMFD level, rather than the reactivity profile. The reason is that in this period, the power growth ratio can be very large. The reactivity varies nearly linearly when the control rods are fully withdrawn from the core. The error predicted here is of the order of 10^{-7}. Then the predicted time step size is of the order of 0.01 s. Since the problem is a super-prompt critical problem with $\beta = 7.2 \times 10^{-3}$, $\Lambda = 2.03 \times 10^{-5}$ and $\rho_{max} = \$1.43236$, the power can be amplified significantly in a single step. The power growth ratio at each step for the case with $\epsilon_m^{in} = 2 \times 10^{-4}$ is shown in Figure 8.4. It can be seen that after $t > 0.05$ s, the power growth ratio increases from order 1 to order 10^4 in 0.2 seconds. This large amplification factor inevitably introduces error in the reactivity estimation. The relative error of the reactivity is also shown in Figure 8.4. The error in the reactivity is relatively small, that is $\sim 5 \times 10^{-5}$. But the problem is a super-prompt critical transient, the error in the power is very sensitive to the error in the reactivity. Therefore for a period of 0.2 s, a relative error of $\sim$0.5% is accumulated.

When the time step size is controlled to limit the power growth ratio, then it can be seen that the relative error for $t > 0.05$ s varies slowly, because the relative error in the reactivity is reduced to $\sim 8 \times 10^{-6}$ suggested by Figure 8.4.

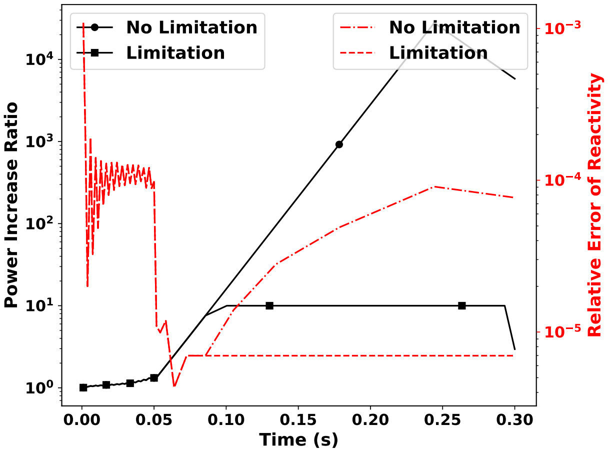


Figure 8.4: Evolution of the power growth ratio in a transient pin problem with the adaptive method to control the time step size. $\epsilon_m^{in} = 2 \times 10^{-4}$.

These results motivate limiting the power growth ratio by illustrating what happens if it is not limited. However, we do not expect that the power increase will be as significant as illustrated for practical simulations due to the presence of the feedback.

In terms of efficiency, it can be seen that all the cases take fewer steps than the case with the default time step. The run time savings are from the fewer time steps performed for $t >$ 0.05 s. For $t <$ 0.05 s, the ATS method does not improve the efficiency. One reason is that the first time size is 1 ms. Therefore more steps will be performed before $\Delta t >$ 5 ms. The other reason is that the variation of cross section introduces relatively large error. To reduce these errors, the method developed here will limit the size of the time step.

8.2.3.2 4-mini Test Problem

Results in Figures 8.5 show the effects of different approaches in calculating the time step size for the 4-mini test problem. The problem is simulated with the 8-group cross section library. Without controlling the one-step error, as suggested by Figure 8.5a, the relative error of the power can change significantly. This is observed at about 0.12 s, 0.25 s and 0.33 s.

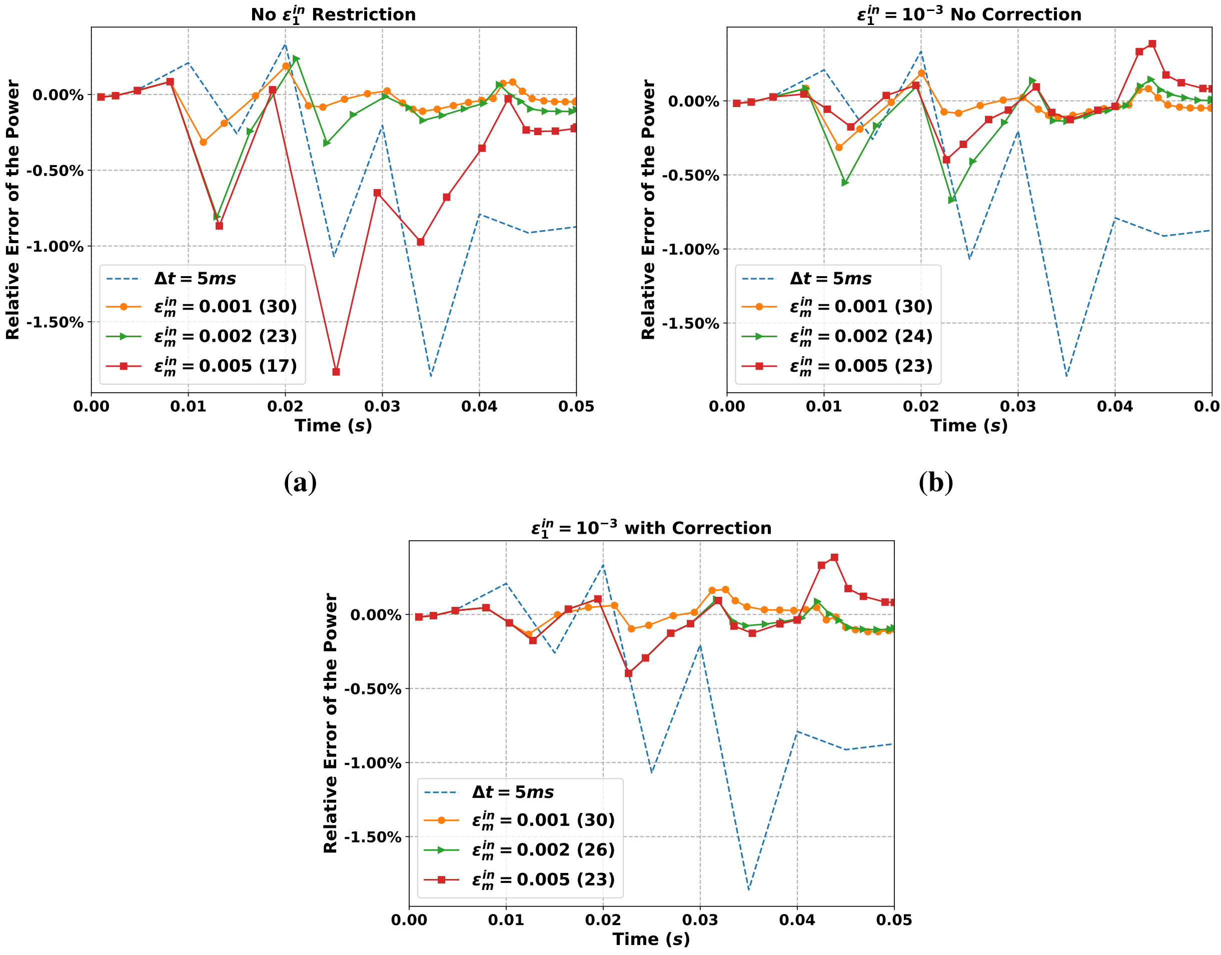
Therefore, the local error must be controlled. Figure 8.5b shows the results when the time step size is also limited to control the one-step error. Then it can be seen that for the cases with $\epsilon_m^{in} = 0.002$ and $\epsilon_m^{in} = 0.005$, the magnitudes of the peaks of the relative error profiles are reduced. When the correction is also applied with the safety factor from Eq. (8.11), as suggested in Figure 8.5c, the magnitudes of the peaks for the relative error profiles are reduced further, and the error evolution becomes smoother.

The results illustrated in Figures 8.5 also show the effect of controlling the multi-step error. As the tolerance, ϵ_m^{in}, becomes smaller, it is observed that the overall relative error in the power becomes smaller.

These approaches, however, reduce the error at the cost of efficiency. For the case with $\epsilon_m^{in} = 0.005$ in Figure 8.5c, $\sim$130 % more steps are performed compared to the default cases. This is a necessary cost, because using the default time step size introduces a 2$\%$ error in the power.

After the control rod movement, the ATS is able to increase the time step size continuously. Figures 8.6 show the evolution of the relative error and time step size for the problem that is simulated for 0.4 s, with $\epsilon_1^{in} = 0.001$. Here it is shown that using the ATS gives power results for $t >$ 0.05 s that have errors smaller than the default results. Moreover, time step sizes become larger than 5 ms in a short time after 0.05 s, and eventually reduces the overall run time of the calculation. The time step size does not exceed 60 ms because of the limitation on the power growth ratio.

As suggested by Figures 8.6 and Figures 8.3, using $\epsilon_m^{in} = 5 \times 10^{-3}$ and $\epsilon_1^{in} = 10^{-3}$ is sufficient to limit the largest absolute values of the relative error to be smaller than 0.5$\%$. This is used as the default option for the ATS method in the numerical results shown later. 0.5$\%$ is a proper tolerance,



(a)

(b)

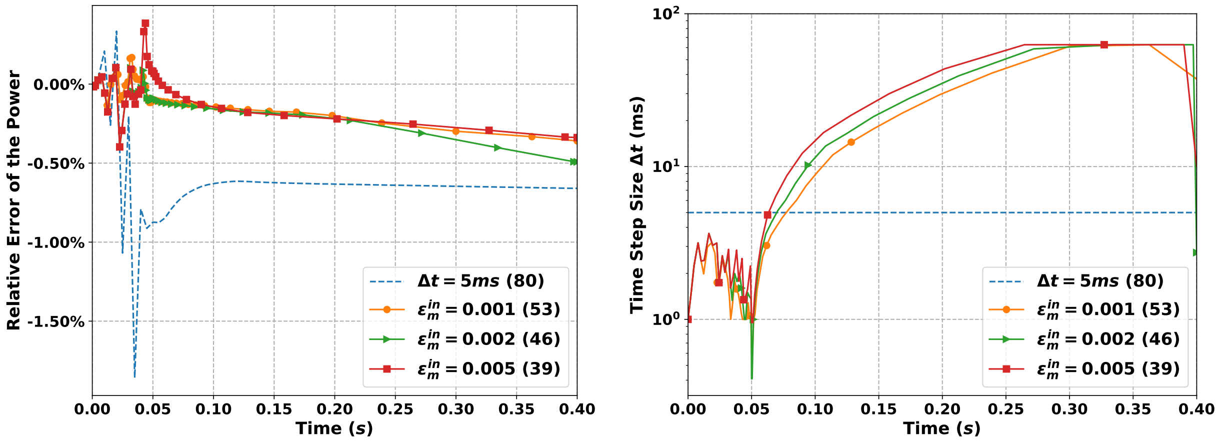
(c)

Figure 8.5: Performance of the ATS method for controlling the internal error of TML in the 4-mini test problem. Only the multi-step error, e_m^{in}, is controlled in (a). The one-step error is also controlled in (b). (c) shows the results with both errors controlled and time step size safety factor correction applied.

since we would like to let the absolute values of the relative error of simulation results smaller than 1%, and the error from coupling has not been considered yet.

8.3 ATS Method for Coupling

As mentioned earlier, the adaptive time-stepping method should also consider the error from OS in addition to the neutronics internal error as these are fundamentally different. In this section, we develop the methodology for estimating and controlling the errors from the OS using a reduced-order model, that is PKE with linear energy feedback (PKE-EF) [88]. Compared to the model



(a) **(b)**

Figure 8.6: Evolution of the relative error and the time step size for different tolerances to control internal error. (a) shows the results for the relative error and (b) shows the results of the time step size. The values in parentheses are the number of step taken for the simulation.

introduced in Section 7.1.3, the heat release constant λ_H, and the initial power are assumed to be zero. Therefore the model is simplified to:

$$\frac{dp(t)}{dt} = \frac{\rho(t) - \beta(t)}{\Lambda(t)} p(t) + \frac{1}{\Lambda_0} \sum_{k=1}^{K} \lambda_k(t)\xi_k(t)\,, \tag{8.17a}$$

$$\frac{d\xi_k(t)}{dt} = \frac{\Lambda_0}{\Lambda(t)} \beta_k(t) p(t) - \lambda_k(t)\xi_k(t)\,,\ k = 1, 2, \cdots K\,, \tag{8.17b}$$

$$\rho(t) = \rho^{ex}(t) + \gamma_d Q(t)\,, \tag{8.17c}$$

$$\frac{dQ(t)}{dt} = p(t)\,. \tag{8.17d}$$

Before presenting the methodology to estimate the error, the PKE-EF model is verified first.

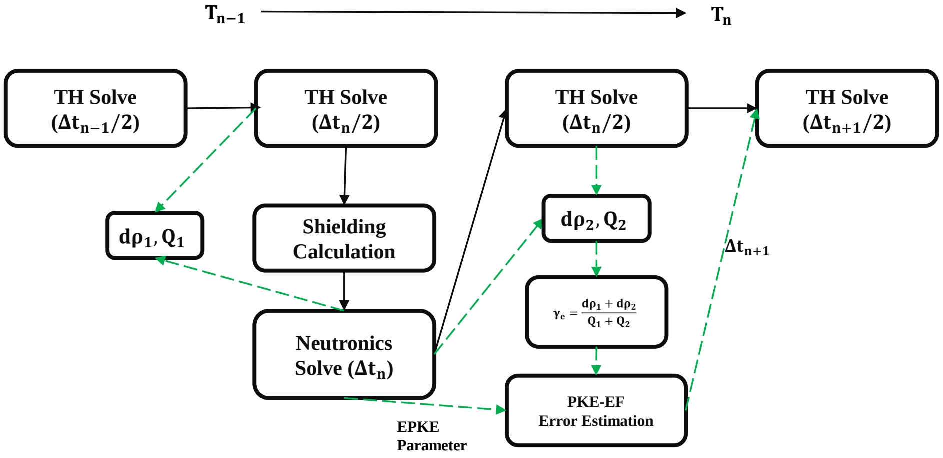
8.3.1 PKE-EF Verification

The PKE-EF model is verified in this section, by showing that the feedback coefficient γ_e is independent of the time step size and that the model is very accurate for predicting the peak power time.

8.3.1.1 Linear Energy Feebdack

This part presents how the γ_e is calculated and shows that the γ_e is independent of the time step size. The flowchart for estimating the feedback coefficient γ_e from the full TML simulation with feedback is shown in Figure 8.7. As indicated by the flowchart, the SS is used here. For each time step, there are two TH calculations that are solved for $\Delta t_n/2$. Supposing that it is the n^{th} step, the energy generation Q_1, and reactivity change, $d\rho_1$ due to the feedback from the first TH solve using $\Delta t_n/2$, is calculated after the first TH calculation. After the NTE is solved with Δt_n, another TH simulation with $\Delta t_n/2$ is performed, and the heat generation Q_2 and reactivity change $d\rho_2$ are calculated. The γ_e is then calculated by

$$\gamma_e = \frac{d\rho_1 + d\rho_2}{Q_1 + Q_2} . \tag{8.18}$$



T_{n-1}
T_n
TH Solve ($\Delta t_{n-1}/2$)
TH Solve ($\Delta t_n/2$)
TH Solve ($\Delta t_n/2$)
TH Solve ($\Delta t_{n+1}/2$)
$d\rho_1, Q_1$
Shielding Calculation
$d\rho_2, Q_2$
Δt_{n+1}
$\gamma_e = \frac{d\rho_1 + d\rho_2}{Q_1 + Q_2}$
Neutronics Solve (Δt_n)
PKE-EF Error Estimation
EPKE Parameter

Figure 8.7: Flowchart of adaptive methodology for coupling.

Since γ_e is introduced to characterize the feedback from TH in a short time, for simulations of the same problem with different time step sizes, the estimated parameter should only depend on the time point and not on the time step size. The properties of γ_e are verified by results from the 4-mini test problem simulated with the 8-group library and shown in Figure 8.8. The initial condition is HZP. It can be seen that, the plots of the estimated γ_e using different time step sizes overlap each other. Therefore, it can be concluded that the model constructed at each time point is independent of time step sizes. It can also be seen in Figure 8.8 that γ_e converge to a near-constant value in a very short time after the reactivity insertion. The value is called the **converged** γ_e. With simulation time increasing, the magnitude of the γ_e becomes smaller because the Doppler feedback coefficient of reactivity becomes more positive with temperature increasing.

The γ_e presented in Figure 8.8 is generated after the reactivity insertion. During the reactivity insertion and the short period after the insertion, the flux shapes change quickly, which makes the estimation of γ_e inaccurate. In such regions, it may be possible to compute a positive γ_e, which is

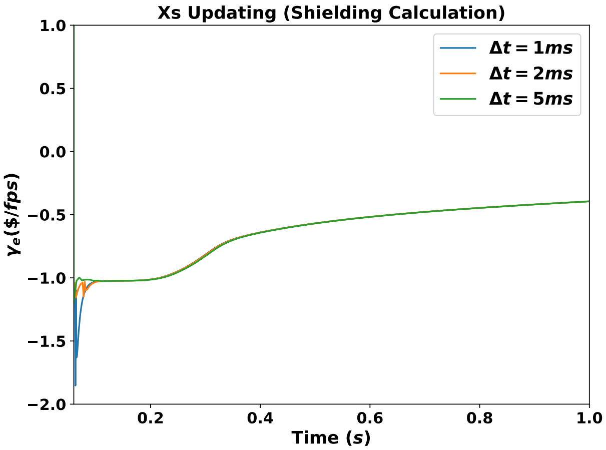


Figure 8.8: Evolution of linear energy feedback coefficient. The nominal power of the model used is more than twice as large as that of the typical reactor, therefore a more negative γ_e is observed compared to that is given in [88].

not physical for a reactor. As a result, if $\gamma_e > 0$, the PKE-EF model becomes invalid, because the system is no longer physically stable. To avoid this issue, we let $\gamma_e = -|\gamma_e|$.

8.3.1.2 Peak Power Time

Another verification of the model is to show the model can predict the time of the peak power for different problems very accurately. The problems investigated here range from 4×4 assemblies to the SMR problem.

We denote the time when the reactivity insertion is complete as t_i. For each problem, the PKE-EF model is built with the converged γ_e and the EPKE parameters at t_i. Due to the presence of feedback, there is a peak power. The predicted time of the peak power, that is denoted by t_p^e, will be compared with the time for the peak power observed in the MPACT simulations, which is denoted by t_p^r. The PKE-EF model is solved by the SDC-IMEX solver. The results from the problems developed base on SPERT assemblies are simulated using the 47-group library and time steps of 0.2 ms. The 4-mini test problem and the 7×7 test problem are simulated with the 51-group library and 1 ms time steps.

The results of these calculations are shown in Table 8.1. It can be seen that the peak power times predicted by the PKE-EF model are very close to the times from the MPACT simulations. For the simple SPERT-2D problems, the predicted time of the peak and the real time of the peak are the same. For the 7×7 test problem, the difference is just 0.3 ms, which is smaller than the time step size used in the MPACT simulations. The case where the PKE-EF model does not predict the peak time so well is the 4-mini-HZP problem. The reason is that the reactivity used is not the maximum reactivity. The maximum reactivity is slightly larger than \$1, and the reactivity at t_i is slightly less than \$1. If the maximum reactivity is used for the PKE-EF model, it can be found that the PKE-EF model can still predict the peak time accurately. Therefore, it can be concluded that PKE-EF model is a reasonably accurate reduced-order model for predicting the evolution of the power.

Table 8.1: Comparison of the peak power times estimated by the PKE-EF model and calculated by MPACT. The reactivity used is the reactivity calculated at t_i. For the 4-mini-HZP problem, the maximum reactivity is also used, and the peak power time is shown in parentheses.

Problem	t_i (s)	Predicted Peak Time t_p^e (s)	Real Peak Time t_p^r (s)
SPERT-2D-HZP	0.05	0.2168	0.2168
SPERT-2D-HFP	0.05	0.0780	0.0780
SPERT-3D-HZP	0.05	0.1396	0.1406
SPERT-3D-HFP	0.05	0.0820	0.0822
4-mini-HZP	0.05	0.8640 (0.7148)	0.723
4-mini-HFP	0.05	0.0784	0.079
7×7 Problem	0.1	0.2937	0.294

8.3.2 Coupling Error Estimation and Adaptive Methodology

Results in Table 8.1 show that the PKE-EF model is accurate to predict the peak power time, indicating that it can capture the feedback efficiently. Therefore, an ATS based on the PKE-EF model with the latest EPKE parameters and γ_e at each time step is proposed.

8.3.2.1 Error Estimation

The ATS method calculates the relative differences in two solutions of the PKE-EF problem. One solution is obtained by solving the PKE-EF problem with the SDC-IMEX method, and is used as the reference. The other solution is obtained by solving the problem with OS.

The relative difference of the two solutions is calculated for a prediction horizon of 0.2 s with time steps of Δt_n. Both the one-step error, e_1^c, and the multi-step error, e_m^c, are obtained. The superscript ^{c} stands for coupling. The algorithm for error estimation is illustrated by Algorithm 17.

8.3.2.2 Evolution of Predicted Error

The evolution of the predicted error in the prediction horizon is investigated in this part. The results are used to show that the effect of coupling error can be understood as the reactivity insertion. The results are also used to explain why LS rather than SS is used for estimating the coupling error for the ATS to control the coupling error.

Algorithm 17 Estimation of the errors to be controlled for coupling.

```
    Input:time step size $\Delta t_n$.
    Output:one-step error $e_1^c$ and multi-step error $e_m^c$
1: Calculate the feedback coefficient $\gamma_e$ by Eq. (8.18).
2: Determine the number of time steps by
       $$N = \min\left(\max\left(20, \lceil \frac{0.2}{\Delta t_n} \rceil\right), 200\right). \quad \text{(Alg 17.1)}$$
3: Solve the PKE-EF using SDC-IMEX solver. The result is $p_r^c(t)$.
4: Solve the PKE-EF with coupling the SDC solver for PKE and direct integration for energy
   deposition $Q$ via LS with $\Delta t_n$. The results are $p_l^c(t)$.
5: Calculate the evolution of the coupling error $e^c(t)$ by
       $$e^c(t) = \left| \frac{p_l^c(t)}{p_r^c(t)} - 1 \right|. \quad \text{(Alg 17.2)}$$
   The one-step error $e_1^c$ is
       $$e_1^c = e^c(\Delta t_n), \quad \text{(Alg 17.3)}$$
   and the maximum of the relative difference in magnitude, $e_m^c$, in the prediction range is
       $$e_m^c = \max_{t \in [0, N\Delta t_n]} e^c(t). \quad \text{(Alg 17.4)}$$
```

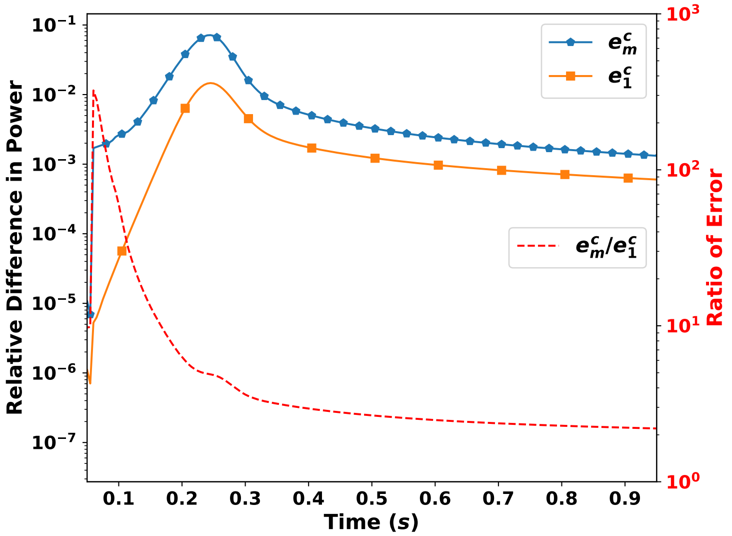


Figure 8.9: Comparison of one-step error and multi-step error.

The ratio of the multi-step error to the one-step error is computed for each time step for the 8-group 4-mini test problem after control rod is fully withdrawn from the core. HZP initial condition is used. The time step size used for error estimation is 5 ms for a period of 0.2 s. So, the PKE-EF model performs 40 steps of calculation as suggested by Eq. (Alg 17.1). It can be observed that for $t \leq 0.1$ s, the ratio

$$A_e = \frac{e_m^c}{e_1^c} >> 40\,. \tag{8.19}$$

While for $t > 0.3$ s, the ratio $A_e \leq 3$.

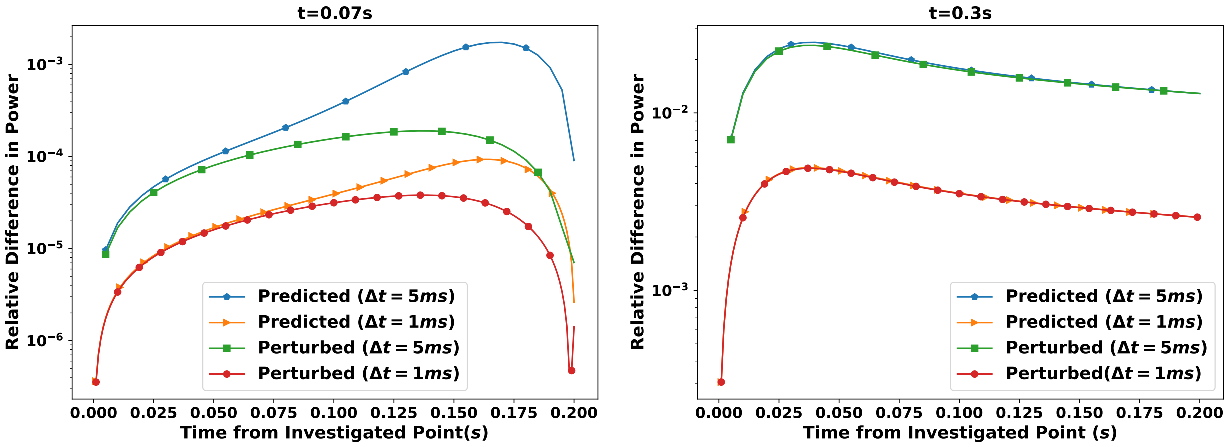
The variation of A_e can be explained from the perspective of reactivity insertion due to the coupling error. The details are shown here. The PKE-EF models at $t = 0.07$ s and $t = 0.3$ s, are perturbed with a reactivity, $\delta\rho^0$, that is defined by

$$\delta\rho^0 = \gamma_e P(t)\frac{\Delta t}{2}\,. \tag{8.20}$$

The model is then solved with the SDC-IMEX method for 0.2 s, and the solution is $p_p^c(t)$. The relative difference $e^p(t)$ is calculated by

$$e^p(t) = \left|\frac{p_p^c(t)}{p_r^c(t)} - 1\right|\,. \tag{8.21}$$

The results are shown in Figures 8.10 (labeled as “Perturbed”). For comparison, the evolution of relative difference calculated by the coupling error estimator, i.e., $e^p(t)$ (defined in Algorithm 17) is shown in Figures 8.10 (labeled as “Predicted”). The error at the first time point for the plots labeled as “Predicted” in Figures 8.10 is the one-step error, and the maximum relative error is the multi-step error calculated by Algorithm 17.



(a) **(b)**

Figure 8.10: Evolution of the relative difference as a function of time. The “Predicted” case denotes the results are the relative differences for solving the PKE-EF with LS. The “Perturbed” case denotes the results are the relative differences from solving the PKE-EF with the SDC-IMEX method but with the initial reactivity perturbation of $(\delta\rho^0 = \gamma_e P\Delta t/2)$.

Figures 8.10 show that when $t = 0.07$ s, the one-step error of the predicted solution and the relative difference of the perturbed PKE-EF model at $t = \Delta t$, $e^p(\Delta t)$, are very close for Δt= 5 ms. And when the time step size is reduced to 1 ms, the evolution of the relative error of the predicted solution and the evolution of the relative difference of the perturbed solution agree for a longer forecast interval. Additionally, for $t = 0.3$ s, the evolutions of the errors agree well with each other for the whole time domain regardless of the time step size. Therefore, we can conclude that using LS for the first step generates the error that induces additional reactivity $\sim \frac{P\Delta t\gamma_e}{2}$. Controlling e^c_m is important, because as suggested above, the effect of coupling error will only show up in a long time.

In the practical simulations, the errors are not only from the OS, but also from the solution methods for the NTE, TH equations and cross sections models, etc. These errors are first-order and are ignored if SS is used for estimating the coupling error. Compared to SS, LS introduces a reactivity of $O(P\Delta t)$ at the first step, that is first-order. Therefore, the error predicted by LS provided an estimation of the error of the same order. It should be noted when P is relatively small, due to the error accumulation, the predicted error multi-step error may not be first-order. When the P is relatively large, the error accumulation from OS compared to the error from the first step becomes less significant. The importance of using LS in error estimation is also shown by the numerical results later.

8.3.2.3 Time Step Size Calculation

Once the errors are predicted from the PKE-EF model, the time step size is calculated. Supposing that the tolerance for the one-step error is ϵ^c_1, the time step size is calculated by:

$$\Delta t^{(1)}_{n+1} = \left(\frac{\epsilon^c_1}{e^c_1}\right)^2 \Delta t_n \,. \tag{8.22}$$

This is derived from the first-order assumption about the local truncation error.

Supposing that the tolerance for the multi-step error is ϵ^c_m, the time step size is calculated by:

$$\Delta t^{(2)}_{n+1} = \left(\frac{\epsilon^c_m}{e^c_m}\right)^q \Delta t_n \,. \tag{8.23}$$

Here q is obtained by direct numerical estimation. This numerical estimate is calculated from the multi-step error using time steps of Δt_n and $2\Delta t_n$. The order of accuracy is then estimated by:

$$q = \frac{\log_{10}\left(\frac{e^c_m(2\Delta t_n)}{e^c_m(\Delta t_n)}\right)}{\log_{10} 2} \,. \tag{8.24}$$

q is estimated because the multi-step error is from accumulation. Numerically calculating q can approximate the order of the mixed error. The time step size applicable is the minimum of the calculated time step sizes, i.e., determined by

$$\Delta t_{n+1}^{(3)} = \min\left(\Delta t_{n+1}^{(1)}, \Delta t_{n+1}^{(2)}\right) . \tag{8.25}$$

The step size variation limit is also applied. The final time step size is

$$\Delta t_{n+1}^{c} = \max\left(1\, ms, \max\left(0.5\Delta t_n, \min\left(\Delta t_{n+1}^{(3)}, 1.2\Delta t_n\right)\right)\right) . \tag{8.26}$$

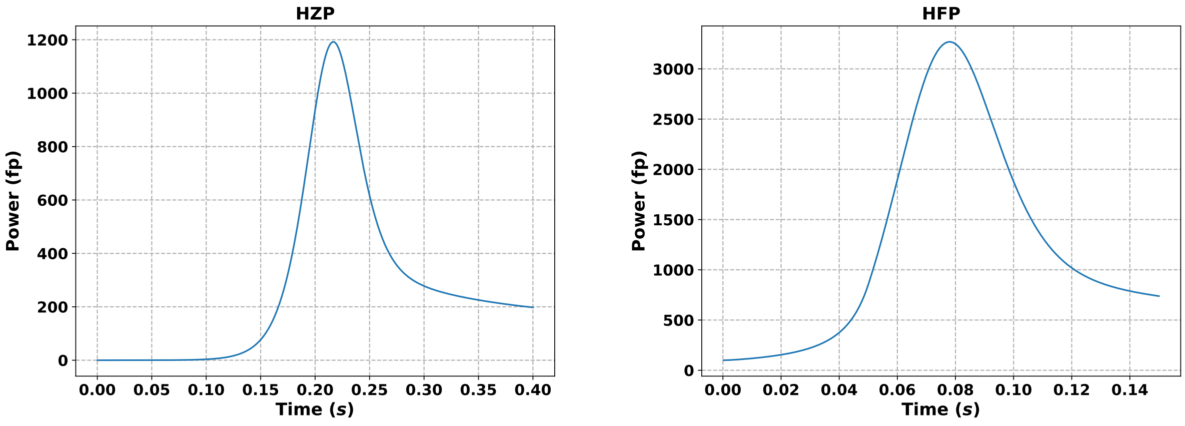
The reason for using a tighter limit on the time step size growth ratio is to reduce the effect of the uncertainty of the γ_e in the short time following the reactivity perturbation. Here, Δt_{n+1}^{c} is the time step size predicted to control the coupling error.

8.3.3 Simple Numerical Results

To demonstrate the efficacy of the ATS method for controlling the OS, some numerical results are analyzed in this section.

The test problems used here are the SPERT-2D problems. The cases from both HZP and HFP conditions are simulated with the evolution of powers shown in Figures 8.11.

These problems are used to show how the tolerances affect the accuracy of the calculation with ATS and provide insights into the tolerances that should be used to estimate the time step size. The goal is to try to make the largest absolute value of the relative error less than 1%.



(a) (b)

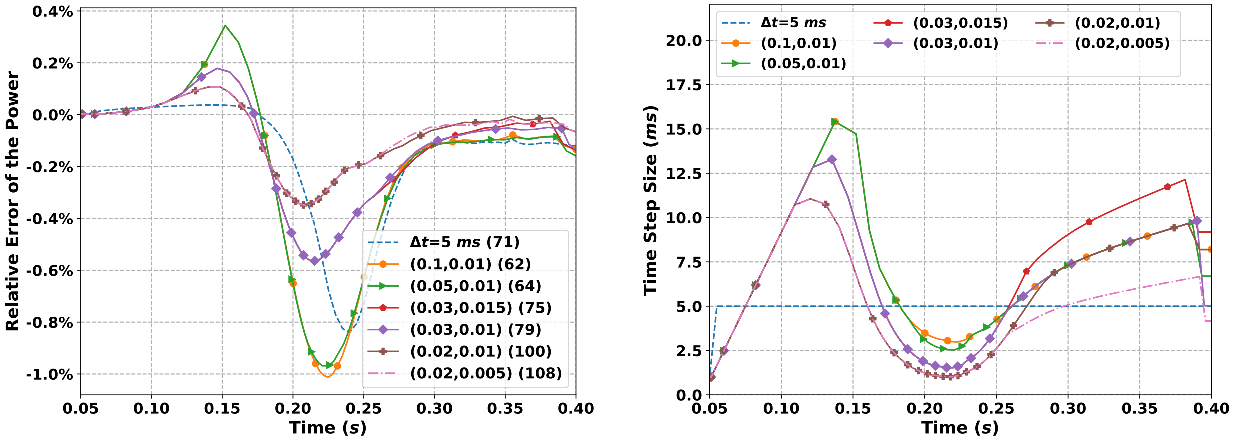
Figure 8.11: Evolution of the power as a function of time for the SPERT-2D cases.

The reference cases are simulated with constant time steps of 0.2 ms. For the reactivity insertion stage ($t < 0.05$ s), all the test cases are simulated with 1 ms. The numerical error from the reactivity insertion is much smaller than the error from the feedback stage. Consequently, the relative error result shown in this section can be considered as the result of using different time step sizes in the feedback stage.

Apart from the test case performed with the default time step of 5 ms for the feedback stage, six additional test cases with ATS are considered and the tolerances used are shown in Table 8.2.

Table 8.2: ATS cases performed in the SPERT-2D problem.

Case	1	2	3	4	5	6
ϵ_m^c	0.1	0.05	0.03	0.03	0.02	0.02
ϵ_1^c	0.01	0.01	0.015	0.01	0.01	0.005



(a)

(b)

Figure 8.12: Relative error of the power and step size as a function of simulation time, for different error tolerances in the SPERT-2D-HZP problem. In the legend, the values in the first parentheses are the tolerances (ϵ_m^c, ϵ_1^c). The value in the second parentheses is the number of time steps.

Figures 8.12 show the performance in terms of the error evolution and the adaptivity of the time steps for the SPERT-2D problem with the HZP condition (SPERT-2D-HZP). First, it is observed with tighter tolerance, the maximum relative error of the power in magnitude becomes smaller. Therefore, the error of the power is consistent with the tolerance. Moreover, the tolerance e_m^c is five times larger than the maximum absolute value of the relative error observed. This indicates that using LS for the error estimation introduces a conservative estimation of the error sources that are not directly estimated.

For the evolution of the time step size, the initial time step size increases to a local maximum. In this phase, the time step size is bounded by the growth ratio 20%. After the maximum, the time step size starts decreasing, because the power is increasing, and the feedback becomes stronger. What is observed is that the time step size changes smoothly, and no oscillation is observed.

During this stage, it can also be observed that ϵ_1^c has no effect on the time step size when $\epsilon_m^c = 0.02$ or $\epsilon_m^c = 0.03$. This indicates that the time step size is mainly determined by controlling the multi-step error. The time step sizes estimated by $\epsilon_m^c = 0.1$ and $\epsilon_m^c = 0.05$ are practically the same. This indicates that the time step size is mainly determined by limiting the one-step error in these two cases. It can also be observed that the time step size reaches its global minimum when the power reaches its maximum. For the asymptotic range, the plots of the evolution of the time step size for cases with the same ϵ_1^c eventually overlap, indicating that the time step size is determined by the ϵ_1^c. So, *controlling ϵ_1^c mainly controls how fast the time step size varies in the asymptotic range.*

Compared to the constant time step size 5 ms, the ATS method predicts a larger time step size when the power is relatively small, and a smaller time step size when the power is relatively large. This agrees with our expectation that to perform a more efficient calculation, the time step should be finer when the power is strong, but can be coarser when the feedback is weaker or the power is in its asymptotic range.

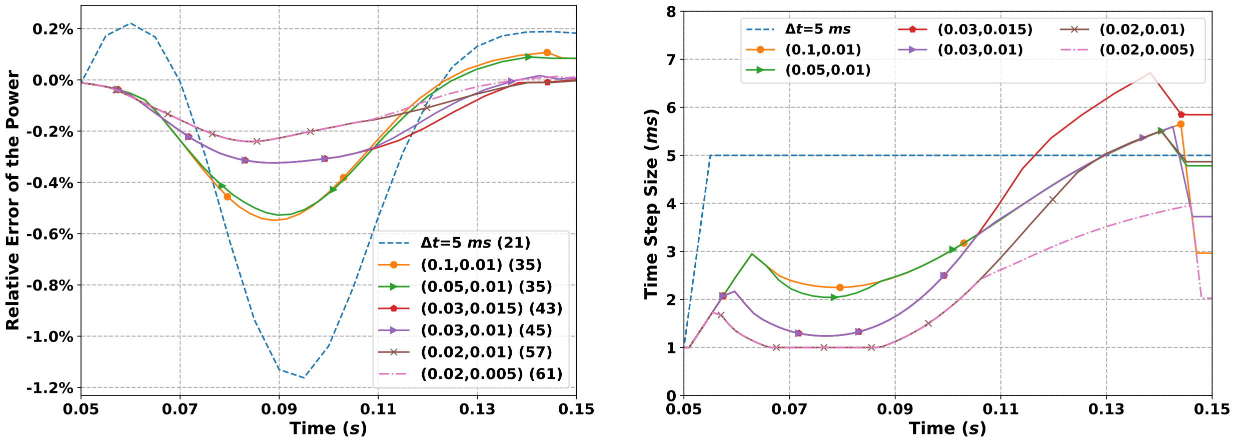
It is noted that when the $\epsilon_m^c = 0.05$ and $\epsilon_m^c = 0.1$, the maximum magnitude of relative error is larger compared to the default case, though the time step size around the peak power is smaller. This result indicates that the time step size for the period before peak cannot be too large, otherwise the error during this period accumulates and gets amplified. The selection of an appropriate ϵ_m^c is very crucial to balance the efficiency and accuracy.

In summary, the results presented here suggest that the ϵ_m^c should be mainly used to limit the maximum magnitude of the relative error, while ϵ_1^c is important to determine the growth rate of the time step size for the asymptotic range. Therefore, in practical simulations, the ϵ_1^c can be large, but ϵ_m^c should not be too large or too small so that balance in the efficiency and accuracy is achieved.

In terms of efficiency, a similar number of steps are performed using $\epsilon_m^c = 0.03$ compared to the default case, but the maximum magnitude of the relative error is reduced by ∼33%. Using the $\epsilon_m^c = 0.02$ results in ∼40% more steps, but reduces the largest absolute value of the relative error from ∼0.8% to ∼0.3%.

Figures 8.13 show the performance of the ATS method in the SPERT-2D with HFP (SPERT-2D-HFP) problem. It is again observed that ϵ_m^c mainly determines the time step size during the pulse. After the peak power, the time step size is eventually determined by the tolerance ϵ_1^c.

Compared to the results for HZP cases, it can be seen that the maximum magnitude of the relative error of the adaptive results is smaller compared to the default case with $\Delta t = 5$ms. However, all the cases perform more time steps. These additional steps are necessary because the default case fails to achieve a result within the error criteria of being less than 1%.



(a) (b)

Figure 8.13: Relative error of the power and time step size as a function of simulation time, for different the error tolerances in the SPERT-2D-HFP case. In the legend, the values in the first parentheses are the tolerances (ϵ_m^c, ϵ_1^c). The value in the second parentheses is the number of time steps.

All the cases indicate using $\epsilon_m^c = 0.02$ and $\epsilon_m^c = 0.03$ can produce the results with the maximum magnitude of the relative error less than 1%. $\epsilon_1^c = 0.01$ or $\epsilon_1^c = 0.015$ generally works fine since the tolerance mainly determines the time step size for the asymptotic stage.

It is also worthwhile to see whether the effectiveness of the tolerances for the adaptive methodology can be predicted by the PKE-EF model. To investigate this, we reproduce the reduced-order PKE-EF problem of the SPERT-2D-HZP test problem. The model is solved by SS. Then we apply the exact same ATS method to the MPACT TML method to determine the time step size for this problem. Recall that in Table 8.1, the PKE-EF model predicted the time of the peak power very close to the MPACT result. The PKE parameters used are obtained at the time the reactivity insertion is complete and the converged γ_e is used for the test problem.

The results are shown in Figure 8.14. It can be seen that for this case, the evolutions of the errors for the MPACT and PKE-EF results are very similar. The maximum magnitudes of the relative errors for the PKE-EF cases are smaller, because the models ignore the errors from solving the NTE and TH equations, and cross section update. In practical simulations, we should not see a similar evolution of the error for the PKE-EF problem and the MPACT simulation though, because the error from the reactivity insertion stage (where control rods are moving) is present for the MPACT results.

However, we believe the predicted accuracy using the adaptive method on the PKE-EF model can be used to determine the right tolerance. On the one hand, the predicted number of steps in the PKE-EF model is very close to the actual number of steps performed in the MPACT simulation.

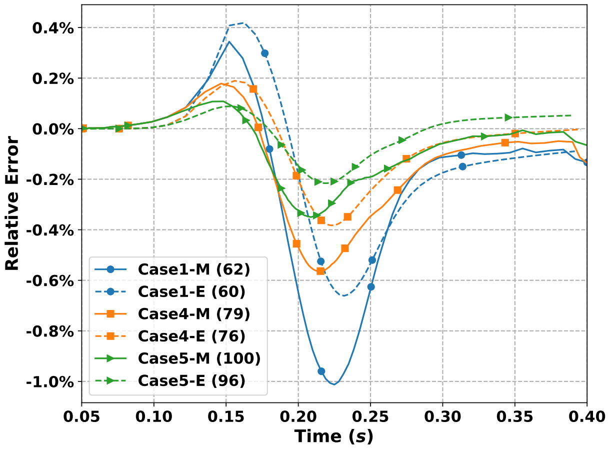


Figure 8.14: Comparison of the performance of the ATS method in the SPERT-2D-HZP cases and the PKE-EF model. "M" in the label stands for MPACT results, and "E" stands for results from the PKE-EF model. The integer shown in parentheses is the number of steps performed. It is observed that the performance of the ATS method to control the coupling error in the SPERT-2D-HZP problem can be predicted by the same method in the PKE-EF problem.

On the other hand, if the maximum magnitude for the PKE-EF model is smaller than 0.5%, then we have some confidence that the results of adaptive simulations using MPACT will not have a relative error larger than 1%.

A future research would be to implement a tolerance predictor based on PKE-EF model. This predictor can adjust the tolerances ϵ_m^c and ϵ_1^c so that the maximum relative error for MPACT results will be smaller than 1%. This predictor can also provide a rough estimation of the number of time steps that needed to be performed. It could be even possible to develop some sophisticated method to plan the time steps in advance, since the PKE-EF model is an accurate reduced-order model capable of predicting the evolution of the power in a short time.

8.4 ATS Flowchart

In this part, we will show the flowchart for the ATS methods. Before showing the flow, the method for the adaptive CMFD level in TML is shown. An adaptive CMFD level is important because, when the predicted time step size is small, it could be unnecessary to use the default number of steps on the CMFD level in the TML method. To reduce the number of CMFD steps and further improve the efficiency, the method to adjust the CMFD level is presented before showing the overall flowchart.

8.4.1 ATS Method for CMFD Level

The approach is developed by investigating the error of using different numbers of CMFD steps compared to the reference results on the 4-mini test problems simulated with time steps of 1 ms, 2 ms and 5 ms. The 4-mini model is the same as the model in Chapter 6 that is simulated with the 51-group library. And the reference solutions are generated with 10 MGCMFD steps per transport step. Since a minimum of 2 MGCMFD steps are required to do quadratic polynomial fitting, the numbers of MGCMFD steps investigated for TML-3 are 2, 3, and 5 (the default). For TML-4, the number of 1GCMFD steps per MGCMFD step is fixed to 3, therefore high-order polynomial fitting can be performed even with 1 MGCMFD step. The numbers of MGCMFD steps investigated for TML-4 are 1, 2, and 3 (the default).

Table 8.3: Accuracy for different CMFD steps per transport step in the 4-mini test problem. The problem is simulated with the 51-group library. The largest absolute values of the relative errors are shown. n_{MG} is the number of MGCMFD steps per transport step. n_{1G} is the number of 1GCMFD steps per MGCMFD step. When n_{1G} is 0, the TML scheme is TML-3, otherwise, it is TML-4.

MOC Step Size	CMFD steps (n_{MG},n_{1G})					
	(2,0)	(3,0)	(5,0)	(1,3)	(2,3)	(3,3)
1 ms	0.06%	0.03%	0.01%	0.12%	0.05%	0.02%
2 ms	0.13%	0.06%	0.02%	0.18%	0.07%	0.04%
5 ms	0.47%	0.21%	0.07%	0.46%	0.13%	0.05%

The results are shown in Table 8.3. The results suggest that to ensure that the maximum of the relative error is smaller than 0.12%, a minimum of 5, 3 or 2 MGCMFD steps should be used for times steps of 1 ms, 2 ms and 5 ms, respectively when TML-3 is used. When TML-4 is used, the minimum number of MGCMFD steps should be 1, 2 or 3 for the investigated time step sizes.

As a result, *the adaptive method for adjusting the number of CMFD steps are proposed as follows:*

$$n_{MG} = \begin{cases} 2 & e_1^{in} < 5 \times 10^{-5} \\ \lfloor \Delta t + 1 \rceil & \Delta t \in [1, 2] \\ \lfloor \frac{2(\Delta t - 2)}{3} + 3 \rceil & \Delta t \in [2, 5] \\ 5 & \Delta t \in [5, \infty) \end{cases} \text{, for TML-3} , \tag{8.27}$$

$$n_{MG} = \begin{cases} 1 & e_1^{in} < 5 \times 10^{-5} \\ \lfloor \frac{\Delta t - 1}{2} + 1 \rceil & \Delta t \in [1, 5] \\ 3 & \Delta t \in [5, \infty) \end{cases} , \; n_{1G} = 3 \text{, for TML-4} .$$

Here the $\lfloor \cdot \rceil$ stands for the nearest integer, and Δt is in units of ms. When the reactivity profile on the MGCMFD level is nearly linear, e_1^{in} is very small, the n_{MG} is also adjusted because there is no need for multiple steps.

It should be noted that the values shown in Table 8.3 are the maximum relative errors. The error accumulated per step from using the n_{MG} calculated by Eq. (8.27) should be much smaller, and will have little effect on the adaptive methodology for controlling the internal TML error. Though a more sophisticated model can be derived to determine the relation between the n_{MG} and the time step size, the model in Eq. (8.27) does provide a reasonably accurate result.

Figure 8.15 compares the results of the adaptive calculation with and without Eq. (8.27) for the 4-mini test problem. It can be seen that using the adaptive CMFD level has little effect on the evolution of the relative error and the ATS methods.

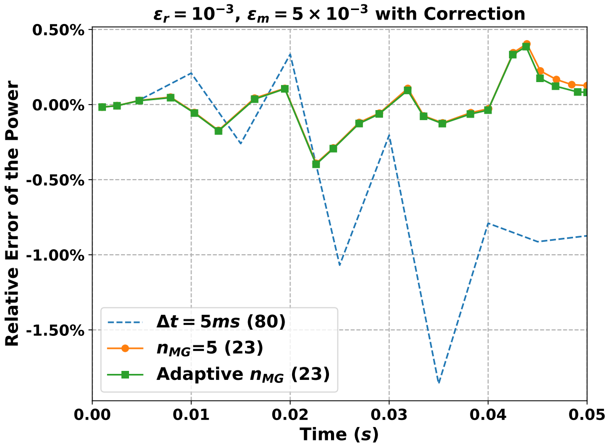
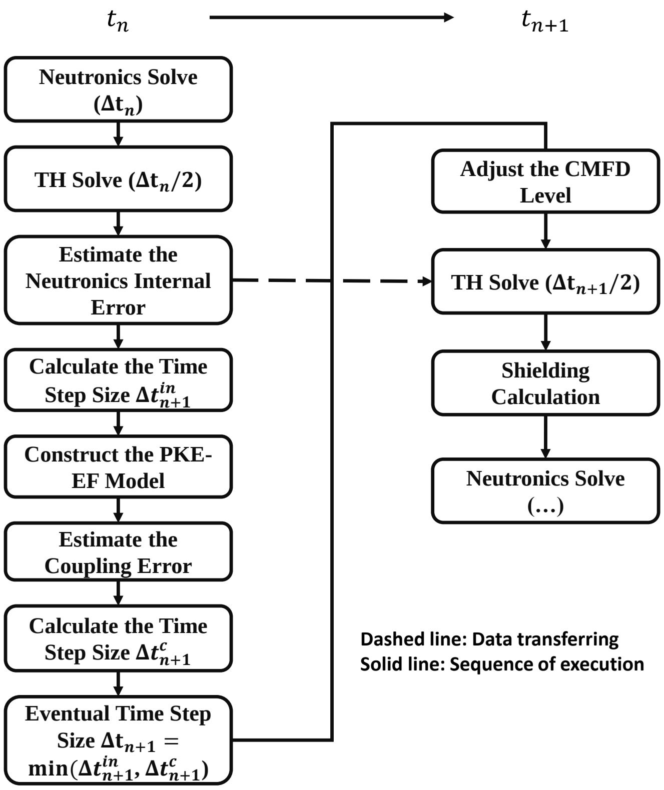


Figure 8.15: Accuracy of adaptive CMFD level in the 4-mini test problem with the 8-group library. The integer shown in parentheses is the number of steps performed.

8.4.2 Overall Flowchart

The overall flowchart for the ATS methodology is presented in Figures 8.16. The error estimation is performed after the second TH solve. The time step size is calculated, and the time step size used for the next time step is the minimum of Δt^{in} and Δt_n^c. The CMFD level of TML for the next time step is adjusted based on the predicted time step size the predicted one-step error using Eq. (8.27). Then the calculation advances to the next step.



t_n
t_{n+1}
Neutronics Solve (Δt_n)
TH Solve ($\Delta t_n/2$)
Estimate the Neutronics Internal Error
Calculate the Time Step Size Δt_{n+1}^{in}
Construct the PKE-EF Model
Estimate the Coupling Error
Calculate the Time Step Size Δt_{n+1}^{c}
Eventual Time Step Size $\Delta t_{n+1} = \min(\Delta t_{n+1}^{in}, \Delta t_{n+1}^{c})$
Adjust the CMFD Level
TH Solve ($\Delta t_{n+1}/2$)
Shielding Calculation
Neutronics Solve (…)
Dashed line: Data transferring
Solid line: Sequence of execution

Figure 8.16: Flowchart of the ATS methods for TML.

8.5 Performance Results

In this section, the performance results of the ATS methods developed in this chapter are verified with different problems.

8.5.1 4-mini Test Problem

The performance of the adaptive methods is first demonstrated for the 4-mini test problems. All the cases use the 51-group library. Four test cases are run. The first case is the TML-4 case utilizing the latest TML scheme in Chapter 7. The other three cases use the ATS methods. The tolerances are shown in Table 8.4. The evolution of power for the 4-mini test problem has been illustrated in Section 6.2.1 is omitted here.

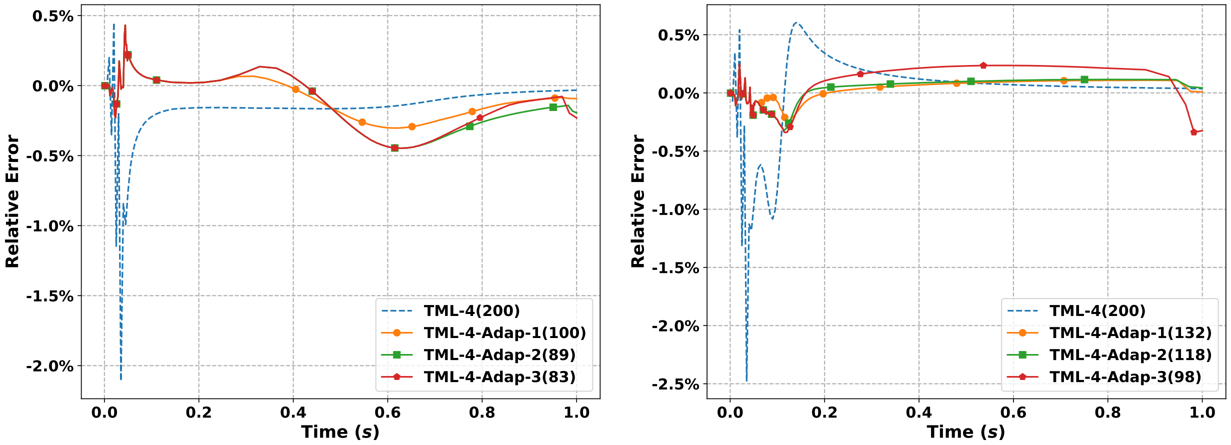
The run time results and the number of time steps performed are presented in Table 8.5. **RR** stands for the run time during the rod movement, where the time step size is mainly determined by the ATS method to control the internal error. **FR** stands for the run time during the feedback stage. **TR** stands for the total run time. Two values are presented for the “Steps”. The first one is

Table 8.4: Tolerances for cases with the ATS methods in the 4-mini test problem.

Case	ϵ_m^{in}	ϵ_1^{in}	ϵ_m^c	ϵ_1^c
TML-4-Adap-1	5×10^{-3}	1×10^{-3}	0.02	0.01
TML-4-Adap-2			0.03	0.01
TML-4-Adap-3			0.03	0.02

Table 8.5: Comparison of adaptive simulations to the simulations with default step size for the 4-mini test problem] The run time results of the TML-3 case are from Table 7.3. The run time results of the TML-4 case are from Figure 7.12a.

Case	HZP				HFP			
	RR (hr)	FR (hr)	TR (hr)	Steps	RR (hr)	FR (hr)	TR (hr)	Steps
TML-3	0.46	4.09	4.55	10/200	0.56	4.41	4.97	10/200
TML-4	0.42	3.43	3.85	10/200	0.42	3.43	3.85	10/200
TML-4-Adap-1	0.75	1.45	2.20	25/100	0.83	2.02	2.85	26/132
TML-4-Adap-2	0.75	1.30	2.05	25/89	0.83	1.77	2.60	26/118
TML-4-Adap-3	0.75	1.20	1.95	25/83	0.83	1.59	2.42	26/98

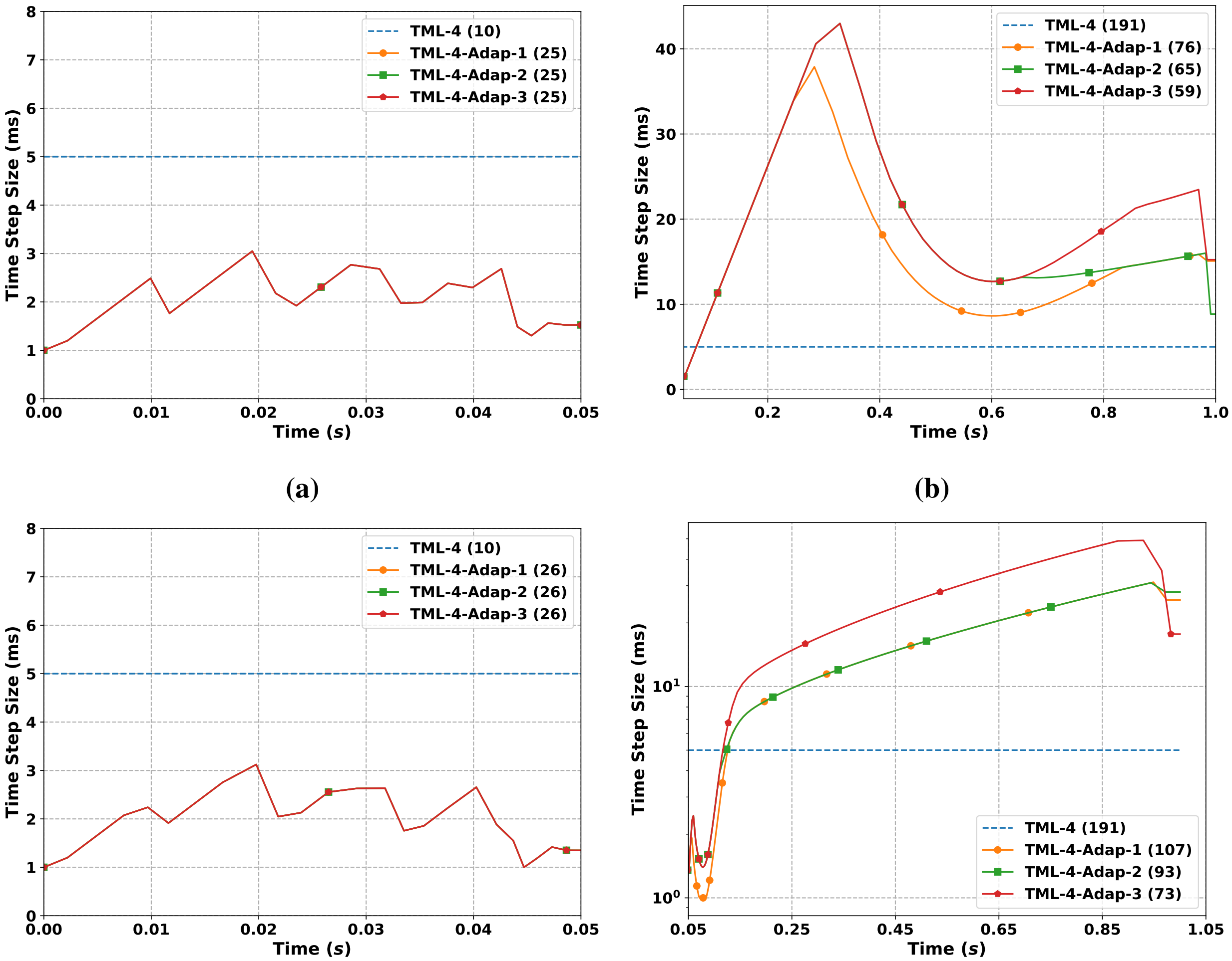


(a)

(b)

Figure 8.17: Relative errors as a function of time in the 4-mini test problems. (a) shows the results for the HZP problem and (b) shows the results for the HFP problem. The values in parentheses are the number of steps performed.

the number of time steps during the rod movement. The second one is the number of time steps for the whole transient process. The evolution of the relative error and the time step sizes are shown in Figures 8.17 and Figures 8.18.



(a) **(b)**

(c) **(d)**

Figure 8.18: Time step size as a function of time in the 4-mini test problems. (a) and (b) show the results for the rod movement period and the feedback stage, for the HZP problem, respectively. (c) and (d) show the results for the HFP problem. The number of time steps performed are also shown in parentheses of the label.

It can be seen that the adaptive method to control the internal error improves the accuracy during the rod movement at the cost of performing more time steps. 2% and 2.5% relative errors in the power results for using the default scheme are observed during the rod movement in the HZP problem and HFP problem, respectively, while the relative error observed from using the ATS methods is less than 0.5%. However, compared to the run time of case TML-4, using the ATS method does cost 80% more time, as suggested by the **RR** results.

The maximum reactivity for the HZP problem is very close to \$1. Therefore, the transient process is relatively slow compared to the super-prompt critical cases. It can be seen that the default 5 ms time step is relatively fine for this problem. The adaptive method to control the coupling error adaptively adjust the time step size. Consequently, though the adaptive cases are not as accurate as the cases with default time step size when the power is relatively large, the largest absolute values of the relative errors of the results for all the cases are still smaller than 1%.

Controlling the error during the reactivity insertion is very important for the simulation of the HFP problem. It can be seen that the peak power result of case TML-4 has a relative error of a magnitude of about 1%, while the absolute values of relative errors for all the cases with the ATS methods are smaller than 0.5%. After the peak power, the time step size for the adaptive cases is increased continuously.

Though more time has been spent during the rod movement stage, for both the HZP problem and HFP problem, the ATS methods reduce the time. Compared to the TML-4, the adaptive methodology can improve the efficiency by at most 50% and 37% for the HZP problem and the HFP problem, respectively. Compared to the TML-3, the run time can be reduced by 57% for the HZP problem, and 50% for the HFP problem.

In terms of the effect of ϵ_1^c and ϵ_m^c, the conclusion made here is the same as what has been observed in Section 8.3.3. We can conclude that ϵ_1^c mainly determines the time step size for the asymptotic stage, and the ϵ_m^c controls the error for the peak power. Both $\epsilon_m^c = 0.02$ and $\epsilon_m^c = 0.03$ help to make the magnitude of the relative error of the power results smaller than 1%, but using $\epsilon_m^c = 0.03$ is more efficient. Using a larger ϵ_1^c can further improve the efficiency.

8.5.2 7×7 Test Problem

The performance of the ATS methods is investigated next for the 7×7 test problem. The reference solution is generated by 1 ms for the first 0.5 s. The peak power is at $t = 0.294$ s, and the asymptotic ranges start at $t \approx 0.4$ s. Therefore, the first 0.5 s is enough to show the evolution of the power. All the test cases are simulated for 0.8 s.

The analyses shown in Chapter 6 and Chapter 7 demonstrate that the maximum absolute value of the relative error from using the 1GCMFD acceleration is smaller than 0.1%. Therefore, for the power results, only the results of the TML-3-1G case are shown.

Two adaptive cases are performed. They use different tolerances for controlling the coupling error. The tolerances are shown in Table 8.6. To control the internal error, all the cases are performed with $\epsilon_1^{in} = 10^{-3}$ and $\epsilon_m^{in} = 5 \times 10^{-3}$. To control the coupling error, the first case is performed with $\epsilon_1^c = 0.01$ and $\epsilon_m^c = 0.02$ and the second case is performed with $\epsilon_1^c = 0.02$ and $\epsilon_m^c = 0.03$. These two cases are performed with TML-4, and the CMFD level is adaptively adjusted during the calculation.

The run time results and the number of time steps taken are shown in Table 8.7. The values shown for steps are the number of time steps in the rod movement stage and the total number of

Table 8.6: Settings for cases with adaptive methods in the 7×7 Problems.

Case	ϵ_m^{in}	ϵ_1^{in}	ϵ_m^c	ϵ_1^c
TML-4-Adap-1	5×10^{-3}	10^{-3}	0.02	0.01
TML-4-Adap-2			0.03	0.02

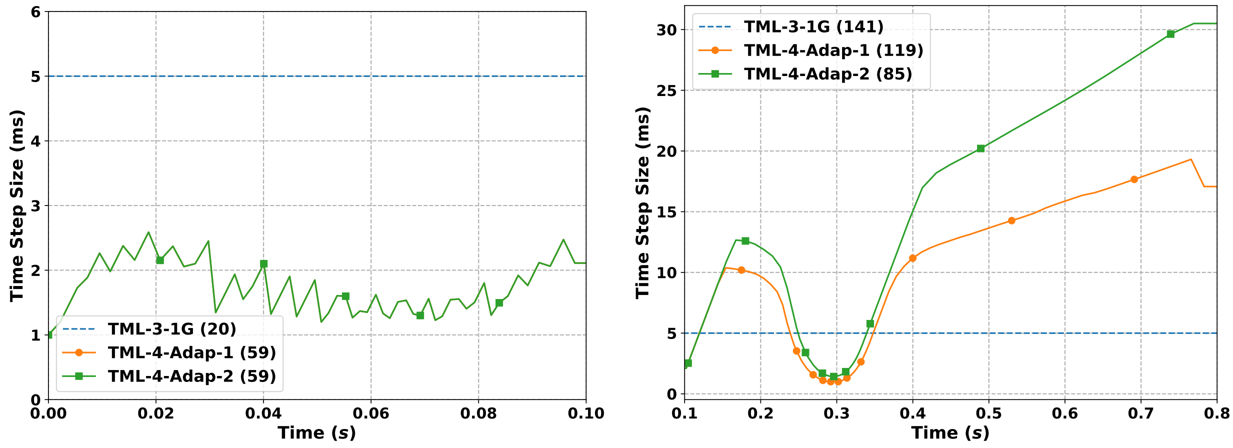
time steps. The evolution of the time step size is shown in Figures 8.19, and the evolution of the relative error is shown in Figure 8.20.

Table 8.7: Comparison of the ATS simulations to the simulations with default step size for the 7×7 test problem. The run time results of case TML-3 are from Table 7.3. The run time results of case TML-4 are from Figure 7.12b.

Case	RR	FR	TR	Steps
	(hr)	(hr)	(hr)	
TML-3	10.3	33.8	44.1	20/160
TML-3-1G	8.2	26.3	34.5	20/160
TML-4	8.0	26.6	34.6	20/160
TML-4-Adap-1	18.0	20.1	38.1	59/177
TML-4-Adap-2	18.0	15.8	33.8	59/144

It can be observed that, compared to the run time of the TML-3 case and the TML-3-1G case, the ATS to control the internal error takes about ∼80% and ∼ 120% more time in the control rod movement stage of the simulation. Compared to the reference case, using the current default time step size results in a maximum error of 6%, while using the ATS methodology reduces the error to 0.7%. Therefore, the results indicate that the default time step size is not fine enough for modeling the rod movement. Also, the time step size keeps oscillating when the control rod is moving because of the heterogeneity along the axial direction.

For the feedback stage of the transient, the run times of ATS cases are much shorter. The time step size is larger than 5 ms when the power is relatively small and is smaller than 5 ms during the time range around the peak power. For the asymptotic range, the time step size continues to increase smoothly as desired. Therefore, it can be observed that fewer time steps are performed. Compared to the TML-3 case, the first ATS case reduces the run time by 41% and the second ATS case reduces the run time by 53%. Compared to the case TML-3-1G, the first ATS case reduces the run time by 23% and the second case reduces the run time by 41%. In terms of accuracy, the relative error does not change initially because the error is dominated again by the rod movement. Immediately after this, the magnitude of the error starts decreasing because of the error cancellation. Eventually, we can see the maximum magnitudes of the relative errors for the



(a) **(b)**

Figure 8.19: Time step size as a function of time in the 7×7 test problem. In the label, the values in parentheses are the number of time steps performed. The results in (a) are the evolution of the time step size during the rod movement. (b) shows the evolution of the time step size for the feedback stage.

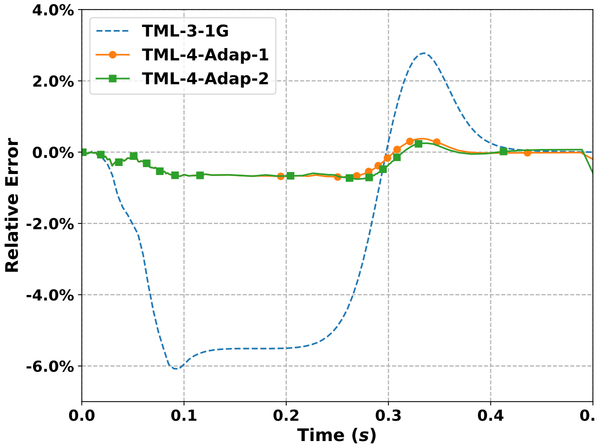


Figure 8.20: Comparison of the relative error of the power for different cases of the 7×7 problem. Compared to the case with default time step size, using the ATS methods developed in the chapter reduced the maximum relative error from 6% to 0.7%.

ATS cases are smaller than 1%. For the TML-3-1G case, due to the error from the rod movement stage, its relative error around the peak power is more than 2%.

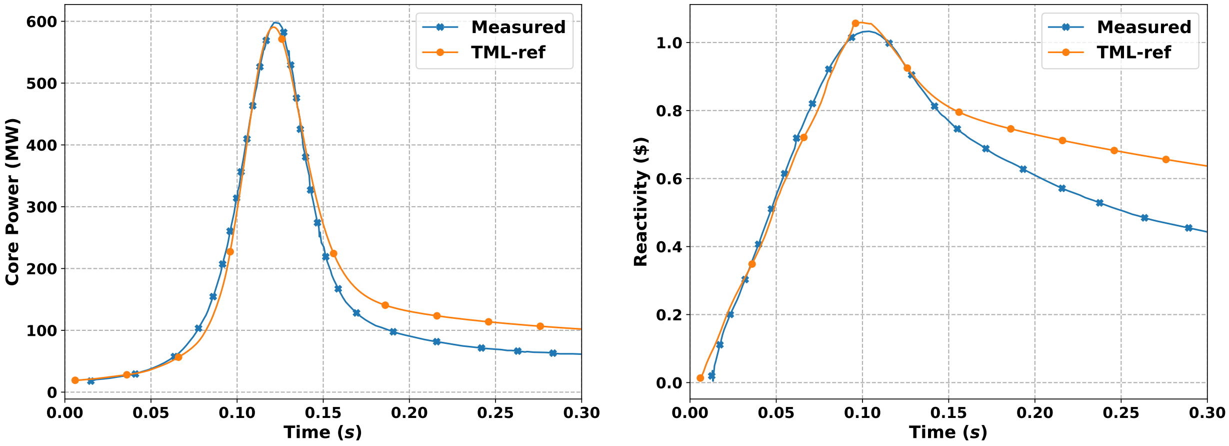
In terms of the error tolerances, the adaptive method to control the coupling using $\epsilon_m^c = 0.02$ or $\epsilon_m^c = 0.03$ yields an efficient simulation overall though.

8.5.3 SPERT Test 86 Problem

The last problem is the SPERT test 86 problem [56]. This is a HFP problem with the initial core inlet temperature of $502\ ^{\circ}F \pm 4\ ^{\circ}F$. The initial TH condition is the same as that in a typical PWR. The initial power is 19 MW, and the withdrawn transient rod worth is $1.17 in 0.105 s. The simulation is performed for 0.3 s.

The SPERT cases are run in MPACT with 0.05 cm ray spacing and the Chebyshev-Gauss quadrature set is used with 4 azimuthal and 1 polar angles. The NEM kernel is used for the axial solution with 20 axial layers. The cross sections are provided from the 51-group library generated at ORNL by CASL [68]. The thermal-hydraulics feedback for the transient solution is determined by the simplified-TH model. The cases were run on the Lighthouse cluster with 20 cores (Xeon E5-2680v3 processor) at the University of Michigan.

Several cases are performed. The reference case is simulated with time steps of 1 ms. As indicated by Figures 8.21, the simulated results of the reference case are in very good agreement with the experiment data [56]. The problem is also simulated with TML-3, TML-3-1G, and the TML-4 with the time steps of the default time step size, 5 ms. For the ATS case, i.e. the case TML-4-Adap, the error tolerances used are $\epsilon_m^{in} = 5 \times 10^{-3}$, $\epsilon_m^{in} = 1 \times 10^{-3}$, $\epsilon_m^{c} = 0.03$, $\epsilon_1^{c} = 0.02$.



(a)

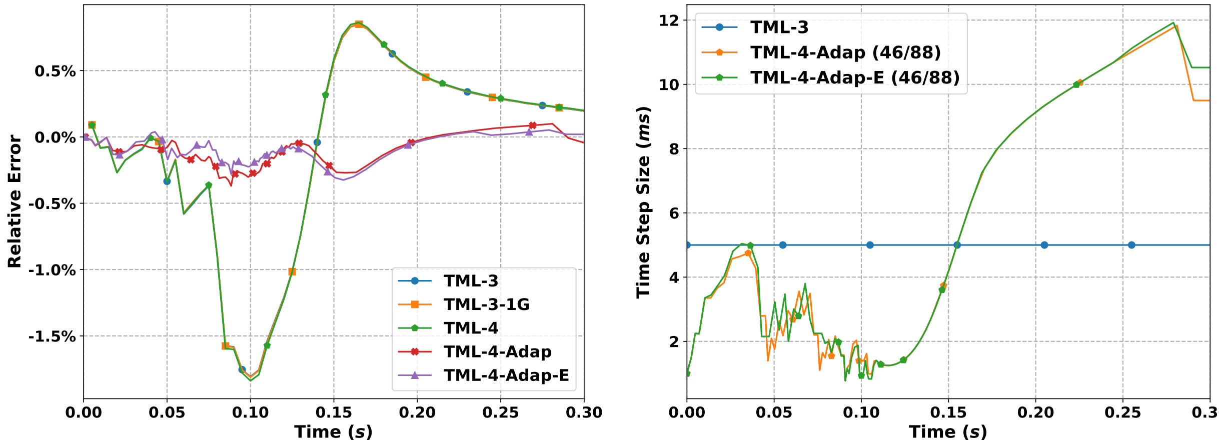
(b)

Figure 8.21: Measured results and the simulated results for the SPERT test 86 problem.

It can be seen that, compared to the default TML-3 simulation, using the 1GCMFD acceleration reduces the run time by more than 24%. Using the ATS methods together with TML-4 can reduce the run time by 22.5%. The case TML-4-Adap performs slower than the case TML-4 because more time steps have been for modeling the rod movement. However, it is necessary to have finer time steps than the default time steps so that the maximum relative error of the power results in magnitude is smaller than 1%. The relative error of the power is shown in Figure 8.22a. It is observed that the maximum absolute value of the relative error for the adaptive calculation is about 0.4%. And using default time step size introduces the error with the maximum magnitude of 1.8%. For the feedback stage, the TML-4-Adap case takes 50% less time to run, though the number of time steps performed is slightly larger than that of TML-4.

Table 8.8: Comparison of adaptive simulations to the simulations with default step size for the SPERT test 86 problem.

Case	RR (hr)	FR (hr)	TR (hr)	Steps
TML-ref	24.2	29.0	53.2	105/300
TML-3	13.0	17.6	30.6	21/60
TML-3-1G	9.9	13.3	23.2	21/60
TML-4	9.4	12.7	22.1	21/60
TML-4-Adap	14.9	8.8	23.7	46/88
TML-4-Adap-E	12.7	8.6	21.3	46/88



(a) **(b)**

Figure 8.22: Comparison of the relative error of the power and time step size for different cases of the SPERT test 86 problem. (a) shows that compared to the default case, using ATS methods developed in the chapter reduced the maximum relative error from 1.8% to 0.4%. (b) shows the evolution of the time step size, with the number of time steps in the rod movement stage and for the whole transient process using the adaptive calculation in parentheses.

The evolution of the time step size is shown in Figure 8.22b. The time step size is smaller in the adaptive calculation for the rod movement stage compared to the default time step size and the time step size reaches its minimum at the peak power. For the asymptotic stage, the time step size keeps increasing smoothly and eventually becomes larger than the default time step size. The simulation is only performed for a transient process of 0.3 s. If a longer transient simulation is performed, the speedup from using the ATS methods will be more significant because the time step size is much larger than 5 ms, during the asymptotic stage.

The ATS case performs more time steps but runs much faster than the TML-4 case after the rod movement. The reason is that the number of CMFD (both MGCMFD and 1GCMFD) steps has been reduced. A new case, TML-4-Adap-E, is performed with TML of 1 MGCMFD step and 2 1GCMFD steps per time step. The results are also shown in Table 8.8 and Figures 8.22. It can be seen the evolution of the relative error for the new case is similar to that of the TML-4-Adap. The maximum magnitude of the error is still ∼0.4%. But the overall run time reduction is ∼30%. Therefore the results indicate the efficiency can be further improved by using the minimum number of CMFD steps required for the ATS methods.

8.6 Summary

In this chapter, the ATS methods based on the PKEs and PKE-EF models for the TML scheme are presented.

The errors from solving the NTE and the coupling are the ones to be controlled. The PKE models are used to estimate the evolution of these two errors in a reasonably long prediction horizon. The ATS methods adjust the time step size based on the error predicted for the next step and the maximum magnitude of the relative error in the prediction horizon, so that these errors do not exceed the user-specified tolerances. Parametric studies were performed to determine the tolerances used that can let the maximum relative error in magnitude of the whole simulation be smaller than 1%.

The effectiveness of the ATS methods was then demonstrated by solving a series of transient test problems, including a 4×4 assembly test problem, a 7×7 problem, and the SPERT test 86 problem with the TML-4 scheme. It was shown that the ATS methods can reduce the total run time by more than 50%, and keep the magnitude relative error of the simulation results smaller than 1% for the 4×4 assembly problem. For the 7×7 test problem, the run time for the period after the reactivity insertion could be reduced by 40%. The method, however, took much more time compared to the default scheme in stage with rod withdrawal. The reason is that the time step size used in the default scheme is relatively coarse. The maximum absolute value of the relative error observed for the default scheme was 6%. Using ATS reduced the error to less than 1%, but spent 80% more time in simulating the rod movement. Nevertheless, using the ATS methods still resulted in a total run time comparable to the TML scheme with 1GCMFD acceleration.

A similar observation was also made in the SPERT test 86 problem. The run time for the period after the reactivity insertion could be reduced by 50%. Using the ATS methods reduced the largest magnitude of the relative error from ∼1.8% to ∼0.4% but spent more time in the rod movement stage. Nevertheless, using the ATS methods together with the TML-4 could reduce the total run time by 22.3% and still take a similar run time to the TML cases with 1GCMFD acceleration using the default time step size.

Therefore, using the ATS methods improves the efficiency for the problem where the default scheme is relatively accurate, and improves the accuracy for the problem where the default scheme is not accurate.

CHAPTER 9

Conclusion & Future Work

9.1 Conclusions

The primary object of this thesis was to develop methods to improve the robustness and the efficiency of time-dependent neutron transport simulations for practical Light Water Reactor (LWR) applications. The first phase of the research here was the investigation of methods to improve the initial condition for the transient. In previous research Multilevel-in-Space-and-Energy Diffusion (MSED), that is an efficient Coarse Mesh Finite Difference (CMFD) solver, was shown to improve the efficiency of the steady-state calculation without feedback, but numerical instabilities occurred with applying the method to the practical coupled physics problem. An original contribution of the research performed here was the development of methods to stabilize MSED for the solution of the coupled physics problem which improved the initial condition for practical transient simulations.

A Fourier analysis was first performed on the CMFD-accelerated scheme coupled with feedback and the partial convergence of low-order diffusion problem was analyzed explicitly in Chapter 3. The Fourier analysis results showed that fully converging the Nonlinear Diffusion Acceleration (NDA) and CMFD equations was very effective for problems without feedback. However, this approach was not adequate for problems with feedback. The Fourier analysis provided a theoretical basis for understanding the reasons why partial convergence of the low-order problem of NDA/CMFD was able to stabilize the multiphysics iteration. Fourier analysis results also predicted that partially converging low-order diffusion problems of NDA/CMFD in problems with feedback can be applied to achieve the same stability as NDA/CMFD in problems without feedback. The result of this research was the development of a relaxation-free iteration scheme and a prescription for the Wielandt shifted ratio and the number of power iterations that were necessary to solve the low-order diffusion at each outer iteration in order to minimize the overall spectral radius. The new method was referred to as the nearly-optimally partially converged CMFD (NOPC-CMFD) method.

In Chapter 4, the NOPC-CMFD method was applied using the CMFD solvers in the Michigan Parallel Characteristics Transport (MPACT) code and numerically verified using practical problems with feedback ranging from a single pressurized water reactor (PWR) fuel rod to a full-core PWR cycle depletion. The results verified the theoretical predictions from Chapter 3, illustrating that the performance of NOPC-CMFD was superior to the existing CMFD or NDA methods with a reduction of up to 40% in the run time for whole-core depletion problems.

In Chapter 5, a more advanced and robust X-CMFD method was also developed using Fourier analysis, and was implemented and optimized in MPACT for practical simulations. The X-CMFD was developed based on the theoretical finding that if the low-order diffusion problem is fully coupled with feedback and solved robustly, the performance of NDA would not be affected by the presence of feedback. The key features of X-CMFD are the application of feedback at the power iteration level of the low-order diffusion problem, and the achievement of stability that is irrespective of the convergence of the CMFD solution and problem feedback intensity. Numerical results verified the stability of the X-CMFD and showed that it provided efficiency comparable to the NOPC-CMFD method when applied to the MSED solver.

Once the stability issue in steady-state simulations was addressed, several methods were developed to improve the efficiency of time-dependent simulations. In Chapter 6, the one-group CMFD (1GCMFD) system was introduced to accelerate the Multigroup CMFD (MGCMFD) solver in TML. A dynamic 1G/MGCMFD iteration strategy was developed, and a new 1GCMFD level was introduced to provide a 4^{th} level to an improvised Transient Multilevel. The new scheme was referred to as TML with one-group CMFD level (TML-4). The 1G/MGCMFD iteration and TML-4 were demonstrated using a series of transport transient kinetics problems in MPACT and were shown to provide improved efficiency. Additionally, TML-4 had better accuracy, except for portions of the transient in which there was control rod movement. Therefore, a hybrid method that used the original TML (TML-3) option with the 1G/MGCMFD iteration when the control rod was moving and TML-4 when there was no control rod movement was proposed. The numerical results showed that the hybrid method was more accurate and that for large-scale problems, the run time of the CMFD solver could be reduced by more than 50%, and the total run time was reduced by as much as 20%.

In Chapter 7, several improvements were made to improve the accuracy of the TML. First, it was found that the accuracy of the TML result was constrained by the accuracy of the exact point-kinetics equations (EPKEs)' solutions. Therefore, the Spectral Deferred Correction (SDC) method was introduced to improve the accuracy of the point-kinetics equation (PKE) method. The implementation of the SDC method was shown to be A-stable for orders up to 8 and the order of accuracy was verified for PKE problems with a range of different reactivities. A 5^{th}-order SDC method was then implemented to solve EPKEs in MPACT, and the error from solutions of the

EPKEs was shown to be negligible. Other improvements to TML in Chapter 7 included modifications to the manner in which feedback was being applied in the solution algorithm. The existing application of TML introduced the feedback at the beginning of the CMFD level implicitly, and at the transport level explicitly via Lie Splitting (LS) in MPACT. The application introduced significant errors when the control rods were modeled explicitly or when there were significant increases or decreases in the time step size. Improvements were therefore investigated in the application of feedback to account not only for explicit control representation, but also for simulations such as adaptive time-stepping which would potentially result in variations in the time step size. The implicit coupling on the CMFD level was removed, and the existing coupling scheme in MPACT based on LS was modified to a method based on Strang Splitting (SS) which made the error consistent with the time step size.

The research in Chapter 7 provided the framework for the investigation of adaptive methods that were presented in Chapter 8 which resulted in one of the most significant and original contributions of the research here in the adaptive time-stepping (ATS) methods. In the ATS methods, two PKE models were investigated to predict the errors for choosing the time step size. One model estimated the error in the neutronics solver assuming no feedback and the other estimated the error within the framework of the coupled solution. The time step size was then adjusted so that errors do not exceed the error tolerances specified. To further improve the efficiency, a method to determine the number of CMFD steps on the CMFD level of TML with respect to the time step size was proposed based on a numerical study. Numerical results showed that when the ATS methods were applied together with TML-4, the run time reduction for the period without control rod movement can be as large as 70%, and for Special Power Excursion Reactor Test (SPERT) test 86 problem, 44%. And the overall time reduction is 22.3%, and the maximum absolute value of the relative error can be reduced from about 1.8% to 0.4% for the SPERT test 86 problem.

9.2 Future Work

9.2.1 Verification of Steady-state Iteration Scheme

The test problems used in this thesis were based on the PWRs and the performance of some of the methods, such as X-CMFD has not been verified and assessed for the whole-core problem. A priority for future work would be to apply X-CMFD to the same whole-core problems that were used to verify the NOPC-CMFD method. The multiphysics simulations of the boiling water reactors (BWRs), for which the physics is more complicated and the feedback is stronger, should then be investigated with the NOPC-CMFD and X-CMFD methods. Extension to other reactor types with very different neutron spectra such as high-temperature gas reactors, fast reactors, and molten salt reactors should then be verified.

The methods derived and demonstrated here assume that the solutions of other physics are fully converged. In practical simulations, equations of other physics are more likely to be solved with iterative methods. In this thesis, when COBRA-TF (CTF) was used to solve the Thermal Hydraulics (TH) equations, the tolerance for converging the CTF solutions was consistent with the fission source residual, but most likely unnecessarily conservative to optimize the performance. Future work should also investigate more sophisticated methods to determine the partial convergence for the solutions of other physics solvers such as CTF.

9.2.2 Improvement on Transient Methodology

Several improvements in the transient methodology are proposed for future work, to include more extensive verification of TML and ATS, as well as high-order methods and the development of multigrid solver for the CMFD Transient Fixed Source Problem (TFSP). This section will briefly outline each of these areas.

9.2.2.1 Verification of ATS and TML

The problems used to verify the ATS methods in this research are super-prompt critical transients in the PWRs with central rod bank ejected. The methods should be verified with other rod banks ejected, as well as a wider range of transients to include sub-prompt critical events. For example, the application of the ATS methods to the simulation of the loss-of-coolant accident should be verified as well as the application of the methods to the BWRs.

In the development of the TML-4 scheme and the adaptive method for adjusting CMFD levels, it was assumed that the solutions with ultra-fine CMFD steps were accurate. This assumption, however, should be verified with additional numerical results. If the introduction of the CMFD level is not sufficiently accurate, then its effectiveness would be compromised for providing a correction to the flux used to calculate the reactivity and an error estimate for adaptive time-stepping.

9.2.2.2 High-order Methods and a Multigrid Solver for CMFD TFSP

Another improvement that can be made in the future is the development and implementation of some high-order methods to solve the time-dependent transport calculation with feedback. The TML scheme is still a first-order method and finer time steps are necessary to further improve the accuracy of the simulations. However, the potential exists for numerical stability issue and significantly higher computational costs as the time step size is refined. Therefore, high-order methods should be considered in the future, which would include the development of an improved

ATS method. It would be important to perform this work within the framework of the coupled neutronics and TH field solution. The ultimate goal in the future could be to solve the time-dependent neutron transport equation (NTE) fully coupled with TH using the high-order methods.

Finally, preliminary numerical results have shown that using 1GCMFD acceleration and a multigrid solver can reduce the run time of the whole-core simulations by more than 50%. However, because of limitations in the accuracy of the solver, the current multigrid method is not suitable for the adjoint problem and CMFD TFSP. An area of future research would be to improve the multigrid solver so that CMFD TFSP can be solved more accurately and efficiently.

APPENDIX A

Fourier Analysis of the Stability of the Steady-state iteration Methods

A.1 Fourier Analysis for Spatially Continuous Problem

In this section, we present the derivation process of the Fourier analysis results shown in Chapter 5 for the spatially continuous problems.

A.1.1 Inner Nonlinear Coupling Term Calculation

To start with, we assume no feedback is present, the low-order diffusion problem is fully converged and the $\hat{D}$ and $\tilde{D}$ is calculated for every inner iteration.

Then the solution of the low-order diffusion problem is written as:

$$\begin{aligned} -\frac{d}{dx} D^{(n,m)}\left(x\right) \frac{d}{dx}\phi^{(n,m+1)}\left(x\right) + \left(\Sigma_a\left(x\right) + \hat{D}^{(n,m)}\left(x\right)\right)\phi^{(n,m+1)}\left(x\right) & \\ = \lambda^{(n,m+1)}\nu\Sigma_f\phi^{(n,m+1)}\left(x\right)\ , & \end{aligned} \tag{A.1}$$

with

$$D^{(n,m)}\left(x\right) = \frac{1}{3\Sigma_t^{(n,m)}\left(x\right)}\ , \tag{A.2a}$$

$$\hat{D}^{(n,m)}\left(x\right) = \frac{\frac{d}{dx}\left[J^{(n)}\left(x\right) + D^{(n,m)}\left(x\right)\frac{d}{dx}\phi^{(n,m)}\left(x\right)\right]}{\phi^{(n,m)}\left(x\right)}\ . \tag{A.2b}$$

To update the transport source, we let $\phi^{\left(n+\frac{1}{2}\right)} = \phi\phi^{\left(n+\frac{1}{2}\right),M}$. Introduce the linearization:

$$\phi^{(n,m)}\left(x\right) = \Phi_0 + \epsilon\phi_1^{(n,m)}\left(x\right)\ , \tag{A.3a}$$

$$J^{(n)}\left(x\right) = \epsilon J_1^{(n)}\left(x\right)\ , \tag{A.3b}$$

$$\phi^{\left(n+\frac{1}{2}\right)}(x) = \Phi_0 + \epsilon\phi_1^{\left(n+\frac{1}{2}\right)}(x)\ , \tag{A.3c}$$

$$\lambda^{(n,m)} = \lambda_0 + O(\epsilon^2)\,. \tag{A.3d}$$

As introduced in Chapter 3, Φ_0 and λ_0 are converged solutions. Substituting Eqs. (A.3) and Eqs. (A.2) into Eq. (A.1), and omitting $O\left(\epsilon^2\right)$ term we have

$$\begin{aligned} -\epsilon\frac{d}{dx}\frac{1}{3\Sigma_{t0}}\frac{d\phi_1^{(n,m+1)}}{dx} &+ \epsilon\Sigma_{a0}\phi_1^{(n,m+1)} \\ &+ \epsilon\left(\frac{dJ_1^{(n)}}{dx} + \frac{d}{dx}\frac{1}{3\Sigma_{t0}}\frac{d\phi_1^{(n,m)}}{dx}\right) = \epsilon\Sigma_{a0}\phi_1^{(n,m+1)}\,. \end{aligned} \tag{A.4}$$

The Fourier Ansatz is introduced by assuming:

$$\phi_1^{(n)}(x) = \theta^n\beta_1 e^{i\omega\Sigma_{t0}x}\,, \tag{A.5a}$$

$$J_1^{(n)}(x) = \theta^n i\beta_2 e^{i\omega\Sigma_{t0}x}\,, \tag{A.5b}$$

$$\phi_1^{(n,m)}(x) = \theta^n\alpha^{(m)} e^{i\omega\Sigma_{t0}x}\,, \tag{A.5c}$$

$$\phi_1^{\left(n+\frac{1}{2}\right)}(x) = \theta^n\beta_3 e^{i\omega\Sigma_{t0}x}\,. \tag{A.5d}$$

Then we obtain

$$\alpha^{(m+1)} = \frac{3}{\omega}\beta_2 + \alpha^{(m)} = \frac{3\left(m+1\right)}{\omega}\beta_2 + \alpha^{(0)}\,. \tag{A.6}$$

Since

$$\alpha^{(0)} = \beta_1\,, \tag{A.7}$$

and the number of inner iterations performed is M. Therefore

$$\beta_3 = \alpha^{(M)} = \frac{3M}{\omega}\beta_2 + \beta_1\,. \tag{A.8}$$

In the Fourier analysis of NDA, we have the relation:

$$\theta\beta_1 = f_{TS}\left(\omega\right)\beta_3\,, \tag{A.9a}$$

$$\theta\beta_2 = -\frac{1}{\omega}\Big[1 - f_{TS}\left(\omega\right)\Big]\beta_3\,. \tag{A.9b}$$

Therefore multiplying Eq. (A.8) by θ, and substituting Eqs. (A.9) into the multiplied equation, we obtain

$$\theta\beta_3 = f_{TS}\left(\omega\right)\beta_3 - \frac{3M}{\omega^2}\Big[1 - f_{TS}\left(\omega\right)\Big]\beta_3\,, \tag{A.10}$$

that is simplified to

$$\theta\left(\omega\right) = f_{TS}\left(\omega\right) - \frac{3M}{\omega^2}\Big[1 - f_{TS}\left(\omega\right)\Big]\,. \tag{A.11}$$

A.1.2 NFCDA

Now, assume that the Nonlinear Fully Coupled Diffusion Acceleration (NFCDA) equation is solved exactly without instability.

The low-order diffusion problem is

$$\begin{aligned} -\frac{d}{dx}D^{\left(n+\frac{1}{2}\right)}\frac{d\phi^{\left(n+\frac{1}{2}\right)}}{dx} + \left[\Sigma_a^{\left(n+\frac{1}{2}\right)}\left(x\right) + \hat{D}^{(n)}\left(x\right)\right]\phi^{\left(n+\frac{1}{2}\right)}\left(x\right) = \\ \lambda^{\left(n+\frac{1}{2}\right)}\left[\nu\Sigma_f^{\left(n+\frac{1}{2}\right)}\left(x\right)\right]\phi^{\left(n+\frac{1}{2}\right)}\left(x\right)\ , \end{aligned} \tag{A.12}$$

with

$$\Sigma_i^{\left(n+\frac{1}{2}\right)}\left(x\right) = \Sigma_{i0}\left(x\right) + \Sigma_{i1}\left(x\right)\left(\phi^{\left(n+\frac{1}{2}\right)}\left(x\right) - \Phi_0\right)\ \ i = t, s, c, \gamma, f\cdots\ , \tag{A.13}$$

and

$$D^{\left(n+\frac{1}{2}\right)}\left(x\right) = \frac{1}{3\Sigma_t^{\left(n+\frac{1}{2}\right)}\left(x\right)}\ , \tag{A.14a}$$

$$\hat{D}^{(n)}\left(x\right) = \frac{\frac{d}{dx}\left[J^{(n)}\left(x\right) + D^{(n)}\left(x\right)\frac{d}{dx}\phi^{(n)}\left(x\right)\right]}{\phi^{(n)}\left(x\right)}\,. \tag{A.14b}$$

Introducing the linearization:

$$\phi^{\left(n+\frac{1}{2}\right)}\left(x\right) = \Phi_0 + \epsilon\phi_1^{\left(n+\frac{1}{2}\right)}\left(x\right)\ , \tag{A.15a}$$

$$J^{(n)}\left(x\right) = \epsilon J_1^{(n)}\left(x\right)\ , \tag{A.15b}$$

$$\lambda^{\left(n+\frac{1}{2}\right)} = \lambda_0 + O(\epsilon^2)\,, \tag{A.15c}$$

then we have the simplified relation,

$$-\frac{d}{dx}\frac{1}{3\Sigma_{t0}}\frac{d\phi_1^{\left(n+\frac{1}{2}\right)}}{dx} + \Sigma_{a1}\Phi_0\phi_1^{\left(n+\frac{1}{2}\right)} + \left(\frac{dJ_1^{(n)}}{dx} + \frac{d}{dx}\frac{1}{3\Sigma_{t0}}\frac{d\phi_1^{(n)}}{dx}\right) = \frac{\Sigma_{a0}\Sigma_{f1}}{\Sigma_{f0}}\Phi_0\phi_1^{\left(n+\frac{1}{2}\right)}\,. \tag{A.16}$$

The Fourier Ansatz is introduced by assuming:

$$\phi_1^{(n)}\left(x\right) = \theta^n\beta_1 e^{i\omega\Sigma_{t0}x}\,, \tag{A.17a}$$

$$J_1^{(n)}(x) = \theta^n i\beta_2 e^{i\omega\Sigma_{t0}x}\,, \tag{A.17b}$$

$$\phi_1^{\left(n+\frac{1}{2}\right)}(x) = \theta^n \beta_3 e^{i\omega\Sigma_{t0}x}\,. \tag{A.17c}$$

We have

$$\frac{\omega^2}{3}\beta_3 - \omega\beta_2 - \frac{\omega^2}{3}\beta_1 = -\gamma\beta_3\,, \tag{A.18}$$

that is simplified to

$$\beta_3 = \frac{\frac{3}{\omega}\beta_2 + \beta_1}{1 + \frac{3\gamma}{\omega^2}}\,. \tag{A.19}$$

We already obtain

$$\theta\beta_1 = f_{TS}(\omega)\left(\beta_3 - \gamma\beta_1\right)\,, \tag{3.18d revisited}$$

$$\theta\beta_2 = -\frac{1}{\omega}\Big[1 - f_{TS}(\omega)\Big]\left(\beta_3 - \gamma\beta_1\right)\,. \tag{3.18e revisited}$$

Therefore solving Eqs. (3.18d), (3.18e) and (A.19) together, we have

$$\theta(\omega) = \frac{f_{NDA}(\omega)}{1 + \frac{3\gamma}{\omega^2}} - \gamma f_{TS}(\omega)\,. \tag{A.21}$$

A.1.3 X-CMFD

X-CMFD is a specific algorithm to solve the fully-coupled low-order diffusion problem. Assuming M inner iterations are performed, the solution of low-order diffusion problem becomes

$$\begin{aligned} -\frac{d}{dx}D^{(n,m)}\frac{d\phi^{(n,m)}}{dx} + \Big[\Sigma_a^{(n,m)}(x) + \hat{D}^{(n)}(x)\Big]\phi^{(n,m)}(x) = \\ \lambda^{(n,m)}\Big[\nu\Sigma_f^{(n,m)}(x)\Big]\phi^{(n,m)}(x)\,, \end{aligned} \tag{A.22}$$

with

$$\Sigma_i^{(n,m)}(x) = \Sigma_{i0}(x) + \Sigma_{i1}(x)\left(\phi^{(n,m-1)}(x) - \Phi_0\right) \quad i = t, s, c, \gamma, f \cdots\,, \tag{A.23}$$

Then we have the simplified relation,

$$\begin{aligned} -\frac{d}{dx}\frac{1}{3\Sigma_{t0}}\frac{d\phi_1^{(n,m)}}{dx} + \Sigma_{a1}\Phi_0\phi_1^{(n,m-1)} + \left(\frac{dJ_1^{(n)}}{dx} + \frac{d}{dx}\frac{1}{3\Sigma_{t0}}\frac{d\phi_1^{(n)}}{dx}\right) = \\ \frac{\Sigma_{a0}\Sigma_{f1}}{\Sigma_{f0}}\Phi_0\phi_1^{(n,m-1)}\,, \end{aligned} \tag{A.24}$$

Introducing the Fourier Ansatz Eqs. (A.5), we have

$$\frac{\omega^2}{3}\alpha^{(m)} - \omega\beta_2 - \frac{\omega^2}{3}\beta_1 = -\gamma\alpha^{(m-1)}\,, \tag{A.25}$$

that is simplified to

$$\alpha^{(m)} = \frac{3}{\omega}\beta_2 + \beta_1 - \frac{3\gamma}{\omega^2}\alpha^{(m-1)}\,. \tag{A.26}$$

When the problem is solved with Wielandt shifted (WS) Power Iteration (PI), then

$$\alpha^{(m)} = \Lambda^L\alpha^{(m-1)} + \left(1-\Lambda^L\right)\left(\frac{3}{\omega}\beta_2 + \beta_1 - \frac{\gamma}{\omega^2}\alpha^{(m-1)}\right)\,. \tag{A.27}$$

The expression is simplified to

$$\alpha^{(m)} = \Big[\left(\Lambda^L - 1\right)\left(\frac{\gamma}{\frac{1}{3}\omega^2}\right) + \Lambda^L\Big]^m \left(\beta_1 - \frac{\frac{3}{\omega}\beta_2 + \beta_1}{\frac{3\gamma}{\omega^2}+1}\right) + \frac{\frac{3}{\omega}\beta_2 + \beta_1}{\frac{3\gamma}{\omega^2}+1}\,. \tag{A.28}$$

Define

$$a = \Big[\left(\Lambda^L - 1\right)\left(\frac{\gamma}{\frac{1}{3}\omega^2}\right) + \Lambda^L\Big]^M\,, \tag{A.29a}$$

$$b = \frac{3\gamma}{\omega^2} + 1\,. \tag{A.29b}$$

Therefore

$$\beta_3 = \alpha^{(M)} = a\left(\beta_1 - \frac{\frac{3}{\omega}\beta_2 + \beta_1}{b}\right) + \frac{\frac{3}{\omega}\beta_2 + \beta_1}{b}\,. \tag{A.30}$$

We already have

$$\theta\beta_1 = f_{TS}\left(\omega\right)\left(\beta_3 - \gamma\beta_1\right)\,, \tag{3.18d revisited}$$

$$\theta\beta_2 = -\frac{1}{\omega}\Big[1 - f_{TS}\left(\omega\right)\Big]\left(\beta_3 - \gamma\beta_1\right)\ . \tag{3.18e revisited}$$

Then we can obtain

$$\theta\left(\omega\right) = \left(a + \frac{1-a}{b} - \gamma\right) f_{TS}\left(\omega\right) - \frac{3}{\omega^2}\frac{1-a}{b}\Big[1 - f_{TS}\left(\omega\right)\Big]\,. \tag{A.32}$$

A.2 Discretized Problem

A.2.1 Partially Converged CMFD

The derivation of the expression for the stability of the partially converged CMFD starts with the discretized form of the neutron transport equation:

$$\mu_p \frac{\psi_{k,p}^{(n)} - \psi_{k-\frac{1}{2},p}^{(n)}}{h} + \Sigma_{t,k}^{(n)} \psi_{k,p}^{(n)} = \frac{\Sigma_{s,k}^{(n)} \phi_k^{\left(n+\frac{1}{2}\right)}}{2} + \frac{\nu \Sigma_{f,k}^{(n)} \phi_k^{\left(n+\frac{1}{2}\right)}}{2} \lambda^{\left(n+\frac{1}{2}\right)} , \qquad \text{(A.33a)}$$

$$1 \leq k \leq K \, , \; 1 \leq p \leq P \, ,$$

where

k = integer index which represents the fine cell center flux,

K = integer which represents the number of fine cells,

$k \pm \frac{1}{2}$ = index which represents the fine cell edge value,

p = integer index which represents the discrete angle,

P = integer which represents the total number of discrete angles,

h = fine cell size.

Closure of the discretized equation is finalized with the auxiliary equations for spatial differencing as defined in Eq. (A.34):

$$\psi_k^{(n)} = \frac{1+\alpha_p}{2} \psi_{k,p}^{(n)} + \frac{1-\alpha_p}{2} \psi_{k-\frac{1}{2},p}^{(n)} . \qquad \text{(A.34)}$$

Here we use

$$\alpha_p = \coth \frac{\Sigma_t h}{2\mu_p} - \frac{2\mu_p}{\Sigma_t h} , \qquad \text{(A.35)}$$

for step-characteristic differencing.

The balance equation of CMFD is written as shown in Eqs. (A.36).

$$\frac{J_{j+\frac{1}{2}}^{\left(n+\frac{1}{2}\right)} - J_{j-\frac{1}{2}}^{\left(n+\frac{1}{2}\right)}}{\Delta_j} + \Sigma_{t,j}^{(n)} \Phi_j^{\left(n+\frac{1}{2}\right)} = \Sigma_{s,j}^{(n)} \Phi_j^{\left(n+\frac{1}{2}\right)} + \lambda^{\left(n+\frac{1}{2}\right)} \nu \Sigma_{f,j}^{(n)} \Phi_j^{\left(n+\frac{1}{2}\right)} , \qquad \text{(A.36a)}$$

$$J_{j+\frac{1}{2}}^{\left(n+\frac{1}{2}\right)} = -D_{j,j+1}^{(n)} \left[\Phi_{j+1}^{\left(n+\frac{1}{2}\right)} - \Phi_j^{\left(n+\frac{1}{2}\right)} \right] + \hat{D}_{j+\frac{1}{2}}^{(n)} \left[\Phi_{j+1}^{\left(n+\frac{1}{2}\right)} + \Phi_j^{\left(n+\frac{1}{2}\right)} \right] , \qquad \text{(A.36b)}$$

of which coefficients are calculated from the homogenization process as:

$$\Phi_j^{(n)} = \frac{1}{q} \sum_{k \in C_j} \phi_k^{(n)} \, , \tag{A.37a}$$

$$J_{j-\frac{1}{2}}^{(n)} = \sum_{p=1}^{P} w_p \mu_p \psi_{q(j-1)+\frac{1}{2},p}^{(n)} \, , \tag{A.37b}$$

$$D_{j,j+1}^{(n)} = \frac{2 D_j^{(n)} D_{j+1}^{(n)}}{\Delta_j D_{j+1}^{(n)} + \Delta_{j+1} D_j^{(n)}} \, , \tag{A.37c}$$

$$\hat{D}_{j+\frac{1}{2}}^{(n)} = \frac{J_{j+\frac{1}{2}}^{(n)} + D_{j,j+1}^{(n)} \left[\Phi_{j+1}^{(n)} - \Phi_j^{(n)} \right]}{\left[\Phi_{j+1}^{(n)} + \Phi_j^{(n)} \right]} \, , \tag{A.37d}$$

$$\Sigma_{i,j}^{(n)} = \frac{\sum_{k \in C_j} \Sigma_{i,k}^{(n)} \phi_k^{\left(n+\frac{1}{2}\right)}}{\Phi_j^{(n)}} \, , \tag{A.37e}$$

where C_j is the range of coarse cell, Δ is the size of the coarse cell, q is the number of fine cells per coarse cell. Eqs. (A.36) are solved with WS power iteration as shown in Algorithm 4. The overall algorithm of the CMFD-accelerated transport iteration is almost the same as Algorithm 8, and is omitted for simplicity. It is assumed that the solution can be linearized by:

$$\psi_{k,p}^{(n)} = \frac{\Phi_0}{2} + \epsilon \tilde{\psi}_{k,p}^{(n)} \, , \tag{A.38a}$$

$$\phi_k^{(n)} = \Phi_0 + \epsilon \tilde{\phi}_k^{(n)} \, , \tag{A.38b}$$

$$J_{j-\frac{1}{2}}^{(n)} = \epsilon \tilde{J}_{j-\frac{1}{2}}^{(n)} \, , \tag{A.38c}$$

$$\Phi_j^{(n)} = \Phi_0 + \epsilon \tilde{\Phi}_j^{(n)} \, , \tag{A.38d}$$

$$\phi_k^{(n)} = \Phi_0 + \epsilon \tilde{\phi}_k^{(n)} \, , \tag{A.38e}$$

$$\Phi_j^{\left(n+\frac{1}{2}\right)} = \Phi_0 + \epsilon \tilde{\Phi}_j^{\left(n+\frac{1}{2}\right)} \, , \tag{A.38f}$$

$$\lambda^{(n+1)} = \frac{\Sigma_{a0}}{\nu \Sigma_{f0}} + O\left(\epsilon^2\right) \, . \tag{A.38g}$$

Eqs. (A.38) are then substituted into Eqs. (A.33) to (A.37).

On the fine mesh, when the linear feedback is applied, the cross section is calculated by:

$$\Sigma_{i,k}^{(n)} = \Sigma_{i0} + \Sigma_{i1} \left(\phi_k^{(n)} - \Phi_0 \right) = \Sigma_{i0,k} + \Sigma_{i1,k} \epsilon \tilde{\phi}_k^{(n)} \, . \tag{A.39}$$

Then the coefficients calculated during the homogenization process is rewritten as:

$$\begin{aligned}\Sigma_{i,j}^{(n)} = \frac{\sum_{k\in C_j}\left(\Sigma_{i0}+\epsilon\Sigma_{i1}\tilde{\phi}_k^{(n)}\right)\left(\Phi_0+\epsilon\tilde{\phi}_k^{(n)}\right)}{\sum_{k\in C_j}\left(\Phi_0+\epsilon\tilde{\phi}_k^{(n)}\right)} &= \Sigma_{i0}+\epsilon\Sigma_{i1}\left(\frac{1}{q}\sum_{k\in C_j}\tilde{\phi}_k^{(n)}\right)\,, \\ &= \Sigma_{i0}+\epsilon\Sigma_{i1}\left(\Phi_j^{(n)}-\Phi_0\right)\,,\end{aligned} \tag{A.40a}$$

$$D_{j,j+1}^{(n)} = \frac{1}{3\Delta\Sigma_{t0}}+\epsilon\tilde{D}_{j,j+1}^{(n)}\,, \tag{A.40b}$$

$$\hat{D}_{j-\frac{1}{2}}^{(n)} = \frac{\epsilon\tilde{J}_{j+1}^{(n)}+\left(\frac{1}{3\Sigma_{t0}}+\epsilon\tilde{D}\right)\frac{\epsilon\left(\tilde{\Phi}_{j+1}^{(n)}-\tilde{\Phi}_j^{(n)}\right)}{\Delta}}{2\Phi_0+\epsilon\left(\tilde{\Phi}_j^{(n)}+\tilde{\Phi}_{j+1}^{(n)}\right)} = \epsilon\frac{1}{2\Phi_0}\tilde{J}_{j-\frac{1}{2}}^{(n)}-\frac{\epsilon\left(\tilde{\Phi}_{j+1}^{(n)}-\tilde{\Phi}_j^{(n)}\right)}{3\Sigma_{t0}\Delta}\,. \tag{A.40c}$$

Therefore Eqs. (A.36) are expressed as:

$$\begin{aligned}&-\frac{\epsilon\left(\tilde{\Phi}_{j+1}^{\left(n+\frac{1}{2}\right)}+\tilde{\Phi}_{j-1}^{\left(n+\frac{1}{2}\right)}-2\tilde{\Phi}_j^{\left(n+\frac{1}{2}\right)}\right)}{3\Sigma_{t0}\Delta^2}+\epsilon\frac{\tilde{J}_{j+\frac{1}{2}}^{(n)}-\tilde{J}_{j-\frac{1}{2}}^{(n)}}{\Delta}+\frac{\epsilon\left(\tilde{\Phi}_{j+1}^{(n)}+\tilde{\Phi}_{j-1}^{(n)}-2\tilde{\Phi}_j^{(n)}\right)}{3\Sigma_{t0}\Delta^2} \\ &+\epsilon\left(\Sigma_{a0}\tilde{\Phi}_j^{\left(n+\frac{1}{2}\right)}+\Sigma_{a1}\tilde{\Phi}_j^{\left(n+\frac{1}{2}\right)}\Phi_0\right)=\epsilon\lambda_0\left(\nu\Sigma_{f0}\tilde{\Phi}_j^{\left(n+\frac{1}{2}\right)}+\nu\Sigma_{f1}\Phi_0\tilde{\Phi}_j^{\left(n+\frac{1}{2}\right)}\right)\,.\end{aligned} \tag{A.41}$$

Cancelling the ϵ term, the error transition of solving the low-order diffusion problem is

$$-\frac{\left(\tilde{\Phi}_{j+1}^{\left(n+\frac{1}{2}\right)}+\tilde{\Phi}_{j-1}^{\left(n+\frac{1}{2}\right)}-2\tilde{\Phi}_j^{\left(n+\frac{1}{2}\right)}\right)}{3\Sigma_{t0}\Delta^2}+\frac{\tilde{J}_{j+\frac{1}{2}}^{(n)}-\tilde{J}_{j-\frac{1}{2}}^{(n)}}{\Delta}+\frac{\left(\tilde{\Phi}_{j+1}^{(n)}+\tilde{\Phi}_{j-1}^{(n)}-2\tilde{\Phi}_j^{(n)}\right)}{3\Sigma_{t0}\Delta^2}+\gamma\Sigma_{t0}\tilde{\Phi}_j^{(n)}=0\,. \tag{A.42}$$

When WS power iteration is applied to solve the CMFD equation, the error transition during the power iteration is

$$\begin{aligned}&-\frac{\left(\tilde{\Phi}_{j+1}^{(n,l+1)}+\tilde{\Phi}_{j-1}^{(n,l+1)}-2\tilde{\Phi}_j^{(n,l+1)}\right)}{3\Sigma_{t0}\Delta^2}+\frac{\tilde{J}_{j}^{(n)}-\tilde{J}_{j-\frac{1}{2}}^{(n)}}{\Delta} \\ &+\frac{\left(\tilde{\Phi}_{j+1}^{(n)}+\tilde{\Phi}_{j-1}^{(n)}-2\tilde{\Phi}_j^{(n)}\right)}{3\Sigma_{t0}\Delta^2}+\gamma\Sigma_{t0}\tilde{\Phi}_j^{(n)}+\left(1-c'\right)\Sigma_{t0}\tilde{\Phi}_j^{(n,l+1)}=\left(1-c'\right)\Sigma_{t0}\tilde{\Phi}_j^{(n,l)}\,,\end{aligned} \tag{A.43}$$

and the error of the solution after L PIs is

$$\tilde{\Phi}_j^{\left(n+\frac{1}{2}\right)}=\tilde{\Phi}_j^{(n,L)}. \tag{A.44}$$

The fine mesh flux is corrected with the coarse mesh solution before the transport sweep. Flat prolongation, linear prolongation and relaxation can be applied. When flat prolongation is applied, the error of the fine mesh flux inside the j^{th} coarse mesh is calculated by:

$$\tilde{\phi}_k^{\left(n+\frac{1}{2}\right)} = \tilde{\phi}_k^{(n)} + \tilde{\Phi}_j^{\left(n+\frac{1}{2}\right)} - \tilde{\Phi}_j^{(n)}\,,\ k \in C_j\,. \tag{A.45}$$

When linear prolongation is used,

$$\begin{aligned}\tilde{\phi}_k^{\left(n+\frac{1}{2}\right)} = \tilde{\phi}_k^{\left(n+\frac{1}{2}\right)} + \frac{1}{2}\left(\tilde{\Phi}_j^{(n)} - \Phi_j^{(n)}\right) + \frac{r-\frac{1}{2}}{2q}\left(\tilde{\Phi}_{j+1}^{\left(n+\frac{1}{2}\right)} - \tilde{\Phi}_{j+\frac{1}{2}}^{(n)}\right) \\ + \frac{q-r+\frac{1}{2}}{2q}\left(\tilde{\Phi}_{j-1}^{\left(n+\frac{1}{2}\right)} - \tilde{\Phi}_{j-1}^{(n)}\right)\,,\ k \in C_j.\end{aligned} \tag{A.46}$$

If an additional under-relaxation factor β is applied, the error of the fine mesh flux is

$$\tilde{\phi}_k^{\left(n+\frac{1}{2}\right)} = \tilde{\phi}_k^{(n)} + \beta\left(\tilde{\phi}_k^{\left(n+\frac{1}{2}\right)} - \tilde{\phi}_k^{(n)}\right). \tag{A.47}$$

With the updated flux, the transport equation of the $O(\epsilon)$ term has the form as:

$$\begin{aligned}&\epsilon\left(\mu_p \frac{\tilde{\psi}_{k+\frac{1}{2},p}^{(n+1)} - \tilde{\psi}_{k-\frac{1}{2},p}^{(n+1)}}{h} + \Sigma_{t0}\tilde{\psi}_{p,k}^{(n+1)} + \Sigma_{t1}\Phi_0\frac{\tilde{\phi}_k^{(n)}}{2}\right) \\ &= \epsilon\left(\Sigma_{t0}\frac{\tilde{\phi}_k^{\left(n+\frac{1}{2}\right)}}{2} + \Sigma_{s1}\Phi_0\frac{\tilde{\phi}_k^{(n)}}{2} + \frac{\Sigma_{a0}}{\Sigma_{f0}}\Sigma_{f1}\Phi_0\frac{\tilde{\phi}_k^{(n)}}{2}\right)\,,\end{aligned} \tag{A.48}$$

which is simplified as:

$$\mu_p \frac{\tilde{\psi}_{p,k+\frac{1}{2}}^{(n+1)} - \tilde{\psi}_{p,k-\frac{1}{2}}^{(n+1)}}{h} + \Sigma_{t0}\tilde{\psi}_{p,k}^{(n+1)} = \Sigma_{t0}\frac{\tilde{\phi}_k^{\left(n+\frac{1}{2}\right)}}{2} - \gamma\Sigma_{t0}\frac{\tilde{\phi}_k^{(n)}}{2}\,. \tag{A.49}$$

Then the errors of the fine mesh scalar flux and the current for the next outer iteration are obtained by

$$\tilde{\phi}_k^{(n+1)} = \sum_{p=1}^{P}\omega_p\tilde{\psi}_{p,k}^{(n)}\,, \tag{A.50a}$$

$$\tilde{J}_{j-\frac{1}{2}}^{(n+1)} = \sum_{p=1}^{P}\omega_p\mu_p\tilde{\psi}_{p,(j-1)q+1}^{(n)}\,. \tag{A.50b}$$

The Fourier Ansatz is

$$\tilde{\psi}_{k,p}^{(n)} = A_{r,p}^{(n)} e^{i\Sigma_{t0}\Delta\omega j} \, , \tag{A.51a}$$

$$\tilde{\Phi}_{j}^{(n)} = C^{(n)} e^{i\Sigma_{t0}\Delta\omega j} \, , \tag{A.51b}$$

$$\tilde{\Phi}_{j}^{(n,l)} = B^{(n,l)} e^{i\Sigma_{t0}\Delta\omega j} \, , \tag{A.51c}$$

$$\tilde{\Phi}_{j}^{\left(n+\frac{1}{2}\right)} = R^{(n)} e^{i\Sigma_{t0}\Delta\omega j} \, , \tag{A.51d}$$

$$\tilde{J}_{j-\frac{1}{2}}^{\left(n+\frac{1}{2}\right)} = G^{(n)} e^{i\Sigma_{t0}\Delta\omega j} \, , \tag{A.51e}$$

$$\tilde{\phi}_{k}^{\left(n+\frac{1}{2}\right)} = F_{r}^{(n)} e^{i\Sigma_{t0}\Delta\omega j} \, , \tag{A.51f}$$

$$\tilde{\phi}_{k}^{(n)} = E_{r}^{(n)} e^{i\Sigma_{t0}\Delta\omega j} \, . \tag{A.51g}$$

Then error transition during homogenization is

$$\mathbf{C}^{(n)} = \frac{\mathbf{1}^{T}}{q}\mathbf{E}^{(n)} \, . \tag{A.52a}$$

When WS power iteration is used, the relation of the errors involved in the CMFD equation is written as:

$$\begin{aligned}\Big[\frac{2-2cos\left(\omega\Sigma_{t0}\Delta\right)}{3\Sigma_{t0}\Delta^{2}} + (1-c')\Sigma_{t0}\Big]B^{(n,l+1)} + \frac{1}{\Delta}G^{(n)}\left(e^{i\omega\Sigma_{t0}\Delta}-1\right) \\ -\Big[\frac{2-2cos\left(\omega\Sigma_{t0}\Delta\right)}{3\Sigma_{t0}\Delta^{2}} - \gamma\Sigma_{t0}\Big]C^{(n)} = \Sigma_{t0}\left(1-c'\right)B^{(n,l)} \, ,\end{aligned} \tag{A.53}$$

with

$$B^{(n,0)} = C^{(n)} \, . \tag{A.54}$$

Define

$$\Theta = \omega\Sigma_{t0}\Delta \, . \tag{A.55}$$

Then the error of the coarse mesh fluxed calculated by CMFD equation is:

$$R^{(n)} = B^{(n,L)} = \Big[\frac{1-c'}{\frac{2-2cos(\Sigma_{t0}\Delta\omega)}{3(\Sigma_{t0}\Delta)^{2}} + 1 - c'}\Big]^{L}\Big[C^{(n)} - \left(\alpha^{(n)} + C^{(n)}\right)\Big] + \left(\alpha^{(n)} + C^{(n)}\right) \, , \tag{A.56}$$

where

$$\alpha^{(n)} = \frac{\frac{-G^{(n)}}{\Sigma_{t0}\Delta}\left(e^{i\Theta}-1\right) - \gamma C^{(n)}}{\frac{2-2cos(\Theta)}{3(\Sigma_{t0}\Delta)^{2}}} = \frac{-G^{(n)}\left(3\Sigma_{t0}\Delta\right)\left(e^{i\Theta}-1\right) - \gamma 3\left(\Sigma_{t0}\Delta\right)^{2}C^{(n)}}{2-2cos\left(\Theta\right)} \, . \tag{A.57}$$

The expression in Eq. (A.56) is eventually simplified as:

$$R^{(n)} = \left(1 - \Lambda^L\right) \alpha^{(n)} + C^{(n)} \,, \tag{A.58}$$

with

$$\Lambda = \frac{1 - c'}{\frac{2 - 2cos(\Sigma_{t0}\Delta\omega)}{3(\Sigma_{t0}\Delta)^2} + 1 - c'} \,. \tag{A.59}$$

Thus the error of the corrected fine mesh flux is:

$$\tilde{\mathbf{F}}^{(n)} = \mathbf{E}^{(n)} + \Big(R^{(n)} - C^{(n)}\Big)\mathbf{u} = \mathbf{E}^{(n)} + \left(1 - \Lambda^L\right) \alpha^{(n)}\mathbf{u} \,. \tag{A.60}$$

$\mathbf{u} \in \mathbb{C}^q$ is the vector characterizing how the solution from the CMFD is used to update the transport solutions. When flat prolongation or linear prolongation, the r^{th} element of $\mathbf{u}$ is defined as:

$$u_r = \begin{cases} 1 \,, & \text{flat prolongation} \,, \\ \frac{2q + (2q - 2r + 1)e^{-i\Theta} + (2r - 1)e^{i\Theta}}{4q} \,, & \text{linear prolongation} \,. \end{cases} \tag{A.61}$$

In matrix form, the errors of the transport solution and the current at the boundary of the coarse mesh are written as:

$$\mathbf{E}^{(n+1)} = \tilde{\mathbf{H}}\Big(\mathbf{F}^{(n)} - \gamma \mathbf{E}^{(n)}\Big) \,, \tag{A.62a}$$

$$G^{(n+1)} = \tilde{\mathbf{G}}\Big(\mathbf{F}^{(n)} - \gamma \mathbf{E}^{(n)}\Big) \,, \tag{A.62b}$$

The definition of $\tilde{\mathbf{H}}$ and $\tilde{\mathbf{G}}$ can be found in [16, 76]. Therefore the error of the fine mesh flux $\mathbf{E}$ and the error of the current G at the boundary of the same coarse mesh are related by:

$$G^{(n)} = \tilde{\mathbf{G}}\tilde{\mathbf{H}}^{-1}\mathbf{E}^{(n)} \,. \tag{A.63}$$

Substituting Eq. (A.60) in Eq. (A.62a), the error of the transport solutions of n^{th} and the next transport solution are correlated in the form as:

$$\begin{aligned} \mathbf{E}^{(n+1)} &= \tilde{\mathbf{H}} \left(1 - \gamma\right) \mathbf{E}^{(n)} + \tilde{\mathbf{H}} \left(1 - \Lambda^L\right) \alpha^{(n)} \mathbf{u} \,, \\ &= \tilde{\mathbf{H}} \left(1 - \gamma\right) \mathbf{E}^{(n)} + \tilde{\mathbf{H}} \left(1 - \Lambda^L\right) \mathbf{u} \frac{-\left(3\Sigma_{t0}\Delta\right)\left(e^{i\Theta} - 1\right)\mathbf{G}^{(n)} - \gamma 3 \left(\Sigma_{t0}\Delta\right)^2 \frac{\mathbf{1}^T}{q} \mathbf{E}^{(n)}}{2 - 2cos\left(\Theta\right)} \,, \\ &= \tilde{\mathbf{H}} \left(1 - \gamma\right) \mathbf{E}^{(n)} + \tilde{\mathbf{H}} \left(1 - \Lambda^L\right) \mathbf{u} \frac{-\left(3\Sigma_{t0}\Delta\right)\left(e^{i\Theta} - 1\right)\tilde{\mathbf{G}}\tilde{\mathbf{H}}^{-1} - \gamma 3 \left(\Sigma_{t0}\Delta\right)^2 \frac{\mathbf{1}^T}{q}}{2 - 2cos\left(\Theta\right)} \mathbf{E}^{(n)} \,. \end{aligned} \tag{A.64}$$

The relation is reformulated as:

$$\tilde{\mathbf{H}}^{-1}\mathbf{E}^{(n+1)} = \Big[\tilde{\mathbf{H}}\left(1-\gamma\right) + \left(1-\Lambda^{L}\right)\mathbf{u}\frac{-\left(3\Sigma_{t0}\Delta\right)\left(e^{i\Theta}-1\right)\tilde{\mathbf{G}} - \gamma 3\left(\Sigma_{t0}\Delta\right)^{2}\frac{\mathbf{1}^{T}}{q}\tilde{\mathbf{H}}}{2-2cos\left(\Theta\right)}\Big]\tilde{\mathbf{H}}^{-1}\mathbf{E}^{(n)}\,. \tag{A.65}$$

Therefore the error transition matrix is

$$\mathbf{T}\left(\omega\right) = \tilde{\mathbf{H}}\left(1-\gamma\right) + \left(1-\Lambda^{L}\right)\mathbf{u}\frac{-\left(3\Sigma_{t0}\Delta\right)\left(e^{i\Theta}-1\right)\tilde{\mathbf{G}} - \gamma 3\left(\Sigma_{t0}\Delta\right)^{2}\frac{\mathbf{1}^{T}}{q}\tilde{\mathbf{H}}}{2-2cos\left(\Theta\right)}\,. \tag{A.66}$$

If the low-order problem is fully converged, then the error of the solution is calculated by

$$\Big[\frac{2-2cos\left(\omega\Sigma_{t0}\Delta\right)}{3\Sigma_{t0}\Delta^{2}}\Big]R^{(n)}+\frac{1}{\Delta}G^{(n)}\left(e^{i\omega\Sigma_{t0}\Delta}-1\right)-\Big[\frac{2-2cos\left(\omega\Sigma_{t0}\Delta\right)}{3\Sigma_{t0}\Delta^{2}}-\gamma\Sigma_{t0}\Big]C^{(n)}=0\,, \tag{A.67}$$

which leads to

$$R^{(n)} = \alpha^{(n)} + C^{(n)}. \tag{A.68}$$

Following the same procedure as shown from Eqs. (A.60) to (A.66), the error transition matrix is

$$\mathbf{T}\left(\omega\right) = \tilde{\mathbf{H}}\left(1-\gamma\right) + \mathbf{u}\frac{-\left(3\Sigma_{t0}\Delta\right)\left(e^{i\Theta}-1\right)\tilde{\mathbf{G}} - \gamma 3\left(\Sigma_{t0}\Delta\right)^{2}\frac{\mathbf{1}^{T}}{q}\tilde{\mathbf{H}}}{2-2cos\left(\Theta\right)}\,. \tag{A.69}$$

When under-relaxation factor β is also used to update the flux, then

$$\tilde{\mathbf{F}}^{(n)} = \mathbf{E}^{(n)} + \Big(R^{(n)} - C^{(n)}\Big)\mathbf{u}\beta\,. \tag{A.70}$$

Following the same procedure as shown in Eqs. (A.60) to (A.66), the error transition matrix is

$$\mathbf{T}\left(\omega\right) = \tilde{\mathbf{H}}\left(1-\gamma\right) - \mathbf{u}\beta\frac{\left(3\Sigma_{t0}\Delta\right)\left(e^{i\Theta}-1\right)\tilde{\mathbf{G}} + \gamma 3\left(\Sigma_{t0}\Delta\right)^{2}\frac{\mathbf{1}^{T}}{q}\tilde{\mathbf{H}}}{2-2cos\left(\Theta\right)}\,. \tag{A.71}$$

A.2.2 X-CMFD

Most of the derivation process for the X-CMFD is the same as the partially converged CMFD. The balance equation of CMFD is written as shown in Appendix A.2.2.

$$\frac{J^{(n,m)}_{j+\frac{1}{2}} - J^{(n,m)}_{j-\frac{1}{2}}}{\Delta_{j}} + \Sigma^{(n)}_{t,j}\Phi^{(n,m)}_{j} = \Sigma^{(n)}_{s,j}\Phi^{(n,m)}_{j} + \lambda^{(n,m)}\nu\Sigma^{(n)}_{f,j}\Phi^{(n,m)}_{j}\,, \tag{A.72a}$$

$$J^{(n,m)}_{j+\frac{1}{2}} = -D^{(n)}_{j,j+1}\left(\Phi^{(n,m)}_{j+1} - \Phi^{(n,m)}_{j}\right) + \hat{D}^{(n)}_{j+\frac{1}{2}}\left(\Phi^{(n,m)}_{j+1} + \Phi^{(n,m)}_{j}\right)\,. \tag{A.72b}$$

The linear discretization is

$$
\begin{aligned}
J_{j-\frac{1}{2}}^{(n,m)} &= \epsilon \tilde{J}_{j-\frac{1}{2}}^{(n,m)} \,, && \text{(A.73a)}\\
\Phi_j^{(n,m)} &= \Phi_0 + \epsilon \tilde{\Phi}_j^{(n,m)} \,, && \text{(A.73b)}\\
\phi_k^{(n,m)} &= \Phi_0 + \epsilon \tilde{\phi}_k^{(n)} \,, && \text{(A.73c)}\\
\Phi_j^{(n)} &= \Phi_0 + \epsilon \tilde{\Phi}_j^{(n,m)} \,, && \text{(A.73d)}\\
\lambda^{(n,m)} &= \frac{\Sigma_{a0}}{\nu \Sigma_{f0}} + O\left(\epsilon^2\right) \,. && \text{(A.73e)}
\end{aligned}
$$

On the fine mesh, when the linear feedback is applied, the cross section is calculated by:

$$
\Sigma_{i,k}^{(n,m)} = \Sigma_{i0} + \Sigma_{i1}\left(\phi_k^{(n,m-1)} - \Phi_0\right) = \Sigma_{i0} + \Sigma_{i1}\epsilon\tilde{\phi}_k^{(n,m-1)} \,. \tag{A.74}
$$

Then the coefficients calculated during the homogenization process is rewritten as:

$$
\begin{aligned}
\Sigma_{i,j}^{(n,m)} &= \frac{\sum_{k\in C_j}\left(\Sigma_{i0} + \epsilon\Sigma_{i1}\tilde{\phi}_k^{(n,m-1)}\right)\left(\Phi_0 + \epsilon\tilde{\phi}_k^{(n,m-1)}\right)}{\sum\left(\Phi_0 + \epsilon\tilde{\phi}_k^{(n,m-1)}\right)}\\
&= \Sigma_{i0} + \epsilon\Sigma_{i1}\left(\frac{1}{q}\sum_{k\in C_j}\tilde{\phi}_k^{(n,m-1)}\right) \,,\\
&= \Sigma_{i0} + \epsilon\Sigma_{i1}\left(\Phi_j^{(n,m-1)} - \Phi_0\right) \,.
\end{aligned} \tag{A.75}
$$

Note that it has been assumed that the transport flux is updated for every inner iteration, and the fine mesh is updated and homogenized to obtain the CMFD coefficient. Therefore Appendix A.2.2 are expressed as:

$$
\begin{aligned}
&-\frac{\epsilon\left(\tilde{\Phi}_{j+1}^{(n,m)} + \tilde{\Phi}_{j-1}^{(n,m)} - 2\tilde{\Phi}_j^{(n,m)}\right)}{3\Sigma_{t0}\Delta^2} + \epsilon\frac{\tilde{J}_{j+\frac{1}{2}}^{(n)} - \tilde{J}_{j-\frac{1}{2}}^{(n)}}{\Delta} + \frac{\epsilon\left(\tilde{\Phi}_{j+1}^{(n)} + \tilde{\Phi}_{j-1}^{(n)} - 2\tilde{\Phi}_j^{(n)}\right)}{3\Sigma_{t0}\Delta^2}\\
&+\epsilon\left(\Sigma_{a0}\tilde{\Phi}_j^{(n,m)} + \Sigma_{a1}\tilde{\Phi}_j^{(n,m-1)}\Phi_0\right) = \epsilon\lambda_0\left(\nu\Sigma_{f0}\tilde{\Phi}_j^{(n,m)} + \nu\Sigma_{f1}\Phi_0\tilde{\Phi}_j^{(n,m-1)}\right) \,.
\end{aligned} \tag{A.76}
$$

Cancelling the ϵ term, the error transition of solving the low-order diffusion problem is

$$
\begin{aligned}
&-\frac{\left(\tilde{\Phi}_{j+1}^{(n,m)} + \tilde{\Phi}_{j-1}^{(n,m)} - 2\tilde{\Phi}_j^{(n,m)}\right)}{3\Sigma_{t0}\Delta^2} + \frac{\tilde{J}_{j+\frac{1}{2}}^{(n)} - \tilde{J}_{j-\frac{1}{2}}^{(n)}}{\Delta}\\
&+\frac{\left(\tilde{\Phi}_{j+1}^{(n)} + \tilde{\Phi}_{j-1}^{(n)} - 2\tilde{\Phi}_j^{(n)}\right)}{3\Sigma_{t0}\Delta^2} + \gamma\Sigma_{t0}\tilde{\Phi}_j^{(n,m-1)} = 0 \,.
\end{aligned} \tag{A.77}
$$

It is assumed that L WS PIs with WS ratio r are performed to solve the low-order diffusion problem for each power iteration. Then

$$\tilde{\Phi}_j^{\left(n+\frac{1}{2}\right)} = \tilde{\Phi}_j^{(n,M,L)}. \tag{A.78}$$

The new Fourier Ansatz is

$$\begin{aligned}
\tilde{\psi}_{k,p}^{(n)} &= A_{r,p}^{(n)} e^{i\Sigma_{t0}\Delta\omega j} \,, \\
\tilde{\Phi}_j^{(n)} &= C^{(n)} e^{i\Sigma_{t0}\Delta\omega j} \,, \\
\tilde{\Phi}_j^{(n,m)} &= B^{(n,m)} e^{i\Sigma_{t0}\Delta\omega j} \,, \\
\tilde{\Phi}_j^{\left(n+\frac{1}{2}\right)} &= R^{(n)} e^{i\Sigma_{t0}\Delta\omega j} \,, \\
\tilde{J}_{j-\frac{1}{2}}^{\left(n+\frac{1}{2}\right)} &= G^{(n)} e^{i\Sigma_{t0}\Delta\omega j} \,, \\
\tilde{\phi}_k^{\left(n+\frac{1}{2}\right)} &= F_r^{(n)} e^{i\Sigma_{t0}\Delta\omega j} \,, \\
\tilde{\phi}_k^{(n)} &= E_r^{(n)} e^{i\Sigma_{t0}\Delta\omega j} \,, \\
\tilde{\Phi}_j^{(n,m,l)} &= B^{(n,m,l)} e^{i\Sigma_{t0}\Delta\omega j} \,.
\end{aligned} \tag{A.79}$$

When WS power iteration is used, the relation of the errors involved in the CMFD equation is written as:

$$\begin{aligned}
&\left[\frac{2-2cos\left(\omega\Sigma_{t0}\Delta\right)}{3\Sigma_{t0}\Delta^2} + (1-c')\Sigma_{t0}\right] B^{(n,m,l+1)} + \frac{1}{\Delta} G^{(n)} \left(e^{i\omega\Sigma_{t0}\Delta} - 1\right) \\
&\qquad - \left[\frac{2-2cos\left(\omega\Sigma_{t0}\Delta\right)}{3\Sigma_{t0}\Delta^2} C^{(n)} - \gamma\Sigma_{t0} B^{(n,m-1)}\right] = \left(1-c'\right)\Sigma_{t0} B^{(n,m,l)} \,,
\end{aligned} \tag{A.80}$$

with

$$B^{(n,m+1)} = B^{(n,m,L)} \,. \tag{A.81}$$

Then the error of the coarse mesh fluxed calculated by CMFD equation for m^{th} inner iteration is:

$$B^{(n,m)} = \left[\left(\Lambda^L - 1\right) \frac{\gamma 3 \left(\Sigma_{t0}\Delta\right)^2}{2-2cos\left(\Theta\right)} + \Lambda^L\right] B^{(n,m-1)} + \left(1-\Lambda^L\right) \left[C^{(n)} - \frac{G^{(n)} \left(3\Sigma_{t0}\Delta\right) \left(e^{i\Theta} - 1\right)}{2-2cos\left(\Theta\right)}\right] . \tag{A.82}$$

After M inner iterations are performed, the error of the CMFD solution is:

$$\begin{aligned}
R^{(n)} = B^{(n,M)} &= a\left(\omega\right) C^{(n)} + \left[1 - a\left(\omega\right)\right] \frac{C^{(n)} - \frac{G^{(n)}\left(3\Sigma_{t0}\Delta\right)\left(e^{i\Theta}-1\right)}{2-2cos(\Theta)}}{1 + \frac{\gamma 3 \left(\Sigma_{t0}\Delta\right)^2}{2-2cos(\Theta)}} \\
&= C^{(n)} + \left[1 - a\left(\omega\right)\right] \frac{-\gamma 3 \left(\Sigma_{t0}\Delta\right)^2 C^{(n)} - G^{(n)} \left(3\Sigma_{t0}\Delta\right) \left(e^{i\Theta} - 1\right)}{2 - 2cos\left(\Theta\right) + \gamma 3 \left(\Sigma_{t0}\Delta\right)^2} \,,
\end{aligned} \tag{A.83}$$

with

$$a(\omega) = \Big[\big(\Lambda^L(\omega) - 1\big) \frac{\gamma 3 \Big(\Sigma_t \Delta\Big)^2}{2 - 2cos\Big(\Sigma_t \Delta \omega\Big)} + \Lambda^L(\omega) \Big]^M . \tag{A.84}$$

Following the same derivation process from Eqs. (A.64) to (A.71), we obtain the error transition matrix:

$$\mathbf{T}(\omega) = \tilde{\mathbf{H}}(1-\gamma) - \Big[1 - a(\omega)\Big] \mathbf{u} \frac{(3\Sigma_{t0}\Delta)\left(e^{i\Theta} - 1\right)\tilde{\mathbf{G}} + \gamma 3 (\Sigma_{t0}\Delta)^2 \frac{\mathbf{1}^T}{q} \tilde{\mathbf{H}}}{2 - 2cos(\Theta) + \gamma 3 (\Sigma_{t0}\Delta)^2} . \tag{A.85}$$

APPENDIX B

Description of Transient Problems

B.1 Pin Test Problem

The pin test problem is developed based on the VERA core physics progression problem 1 in [98]. The fuel material is initially UO_2 with 3.1% enrichment.

When the transient problem is simulated with TH, the rated power of the fuel is 179 W, and the initial power is 100% fp.

For the pin problem in Figures 7.6, the transient process starts with changing the enrichment from 3.1% to 3.3% in 0.025 s. The maximum reactivity is $1.98.

For the pin problem in Figures 7.11, the transient process starts with changing the enrichment from 3.1% to 3.2% in 0.025 s. And the maximum reactivity is $1.0278.

For the pin problem with maximum reactivity of $1.43 in Figure 8.2, the transient process starts with changing the enrichment from 3.1% to 3.24% in 0.05 s.

For the pin problem with maximum reactivity of $0.53 in Figure 8.2, the transient process starts with changing the enrichment from 3.1% to 3.15% in 0.05 s.

B.2 SPERT-2D Test Problem

SPERT-2D test problem is a 2D layer of a 4×4 mini core problem that models 16 fuel assemblies with quarter-core geometry. The layout of the core is shown in Figure B.1. The model consists of four fuel assemblies, which include two 5×5 assemblies, one control rod follower 4×4 assembly, and one transient 4×4 assembly. The transient assembly locates at the lower right corner in Figure B.1 and contains one quarter of the cruciform transient rod composed of 1.35 wt% borated steel. All the assemblies are from SPERT III E-Core experiments and the modeling of their geometries, materials can be found in [55, 92]. The thickness of the layer is 6.942667 cm and the rated power (full power) is 200 kW.

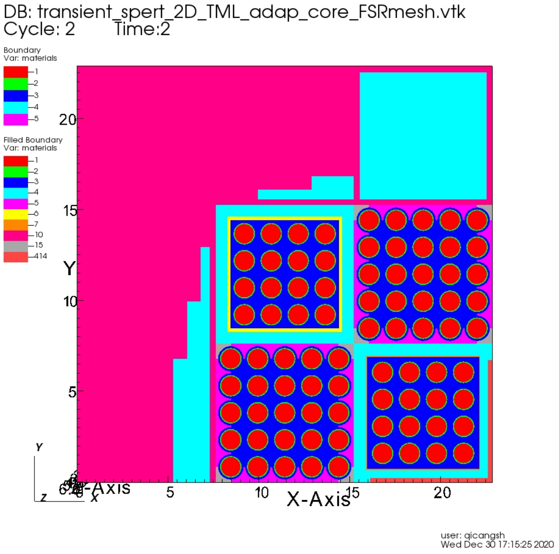


Figure B.1: Layout of the SPERT-2D test problem.

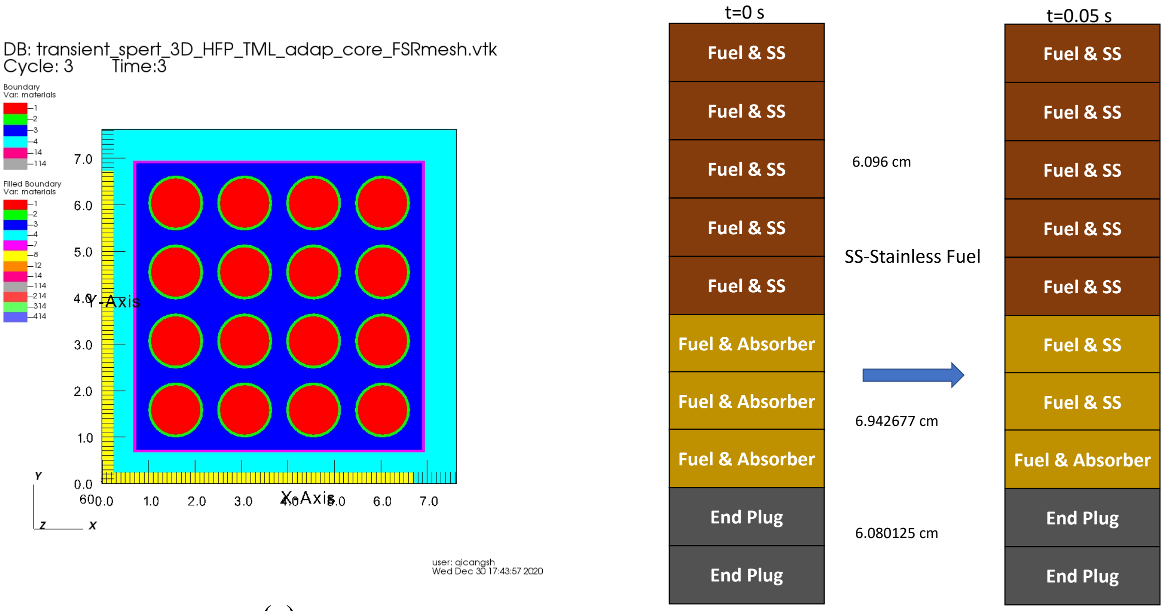
The transient process is induced by withdrawing the rod is out of the core in 0.05 s. The reactivity inserted is around \$1.10. For the modeling, the transient rod is replaced with stainless steel. The initial power is 0.01% full power for the HZP cases, and is 100% for the HFP cases. The inlet temperature is 533.15 K. The simplified-TH is used for modeling [89]. The problem is simulated with the 47-group library.

B.3 SPERT-3D Test Problem

SPERT-3D test problem is a 3D transient assembly from SPERT III E-core. The problem is developed based on the stripped assembly case [54]. The radial geometry is shown in Figures B.2.

The bottom two layers are end-plug and no fuel is present. Above the end-plug are the three assemblies inserted with a transient blade. The absorber for the two assemblies is simultaneously replaced with stainless steel in 0.05 s.

The rated power of the problem is 100 kW. The initial power is 0.01% full power for the HZP cases, and is 100% for the HFP cases. The reactivity inserted is \$1.37. The inlet temperature is 533.15 K. The problem is simulated with the 47-group library. The simplified-TH is used for modeling [89]. All the boundaries are reflective, except that the bottom boundary is the vacuum.



(a)

(b)

Figure B.2: Layout of the SPERT-3D test problem. (a) is the 2D geometry. (b) is the axial layer layout.

BIBLIOGRAPHY

[1] Chen, J., Liu, Z., Zhao, C., He, Q., Zu, T., Cao, L., and Wu, H., "A New High-Fidelity Neutronics Code NECP-X," *Annals of Nuclear Energy*, Vol. 116, June 2018, pp. 417–428.

[2] Joo, H. G., Cho, J. Y., Kim, K. S., Lee, C. C., and Zee, S. Q., "Methods and Performance of a Three-Dimensional Whole-Core Transport Code DeCART," *Proceedings of the PHYSOR 2004: The Physics of Fuel Cycles and Advanced Nuclear Systems - Global Developments*, American Nuclear Society, Chicago, IL, 2004, pp. 21–34.

[3] Jung, Y. S., Shim, C. B., Lim, C. H., and Joo, H. G., "Practical Numerical Reactor Employing Direct Whole Core Neutron Transport and Subchannel Thermal/Hydraulic Solvers," *Annals of Nuclear Energy*, Vol. 62, Dec. 2013, pp. 357–374.

[4] Kochunas, B., Collins, B., Stimpson, S., Salko, R., Jabaay, D., Graham, A., Liu, Y., Kim, K. S., Wieselquist, W., Godfrey, A., Clarno, K., Palmtag, S., Downar, T., and Gehin, J., "VERA Core Simulator Methodology for Pressurized Water Reactor Cycle Depletion," *Nuclear Science and Engineering*, Vol. 185, No. 1, 2017, pp. 217–231.

[5] Weber, D. P., Sofu, T., Yang, W. S., Downar, T. J., Thomas, J. W., Zhong, Z., Cho, J. Y., Kim, K. S., Chun, T. H., Joo, H. G., and Kim, C. H., "High-Fidelity Light Water Reactor Analysis with the Numerical Nuclear Reactor," *Nuclear Science and Engineering*, Vol. 155, No. 3, March 2007, pp. 395–408.

[6] Core, U. S. N. R. C., Simulator, N., Xu, Y., Seker, V., Arbor, A., Hudson, N., and Nrc, R. E. S. U. S., "PARCS v3.0 U.S. NRC Core Neutronics Simulator THEORY MANUAL t." Tech. rep., Depart of Nuclear Engineering and Radiological Sciences, University of Michigan, Ann Arbor, MI 48109, 2009.

[7] Marin-Lafleche, A., Smith, M. A., and Lee, C., "Proteus-MOC: A 3D Deterministic Solver Incorporating 2D Method of Characteristics," *Proceedings of International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C 2013)*, Vol. 4, American Nuclear Society, Sun Valley, ID, United states, 2013, pp. 2759–2770.

[8] Jung, Y. S. and Joo, H. G., "Decoupled Planar MOC Soluition for Dynamic Group Constant Generation in Direct Three-Dimensional Core Calculations," *American Nuclear Society - International Conference on Mathematics, Computational Methods and Reactor Physics 2009, m and c 2009*, Vol. 4, American Nuclear Society, Saratoga Springs, New York, 2009, pp. 2157–2167.

[9] Brown, F. B., "MCNP-a General Monte Carlo n-Particle Transport Code, Version 5 Report LA-UR-03-1987," *Los Alamos, NM: Los Alamos National Laboratory*, Los Alamos National Laboratory, Oak Ridge, TN, 2003.

[10] Romano, P. K. and Forget, B., "The OpenMC Monte Carlo Particle Transport Code," *Annals of Nuclear Energy*, Vol. 51, 2013, pp. 274–281.

[11] Smith, K., "Nodal Method Storage Reduction by Nonlinear Iteration," *Transactions of the American Nuclear Society*, Vol. 44, 1983.

[12] Smith, K. and Rhodes, J., "Full-Core, 2-D, LWR Core Calculations with CASMO-4E," *Proceedings of PHYSOR2002: International Conference on the New Frontiers of Nuclear Technology Reactor Physics, Safety and High-Performance Computing*, No. 1, American Nuclear Society, American Nuclear Society, Seoul, Korea, 2002, pp. 7–10.

[13] Alcouffe, R. E., "Diffusion Synthetic Acceleration Methods for the Diamond-Differenced Discrete-Ordinates Equations," *Nuclear Science and Engineering*, Vol. 64, No. 2, Oct. 1977, pp. 344–355.

[14] Cho, N. Z., Lee, G. S., and Park, C. J., "Partial Current-Based CMFD Acceleration of the 2D/1D Fusion Method for 3D Whole-Core Transport Calculations," *Transactions of the American Nuclear Society*, Vol. 88, 2003.

[15] Li, L., Smith, K., and Forget, B., "Techniques for Stabilizing Coarse-Mesh Finite Difference (CMFD) in Methods of Characteristics (MOC)," *Supercomputing in Nuclear Applications*, Vol. 1, Nashville, TN, 2015, pp. 480–493.

[16] Zhu, A., Jarrett, M., Xu, Y., Kochunas, B., Larsen, E., and Downar, T., "An Optimally Diffusive Coarse Mesh Finite Difference Method to Accelerate Neutron Transport Calculations," *Annals of Nuclear Energy*, Vol. 95, Sept. 2016, pp. 116–124.

[17] Yee, B. C., Kochunas, B., and Larsen, E. W., "A Multilevel in Space and Energy Solver for Multigroup Diffusion Eigenvalue Problems," *Nuclear Engineering and Technology*, Vol. 49, No. 6, Sept. 2017, pp. 1125–1134.

[18] Wang, D. and Xiao, S., "A Linear Prolongation Approach to Stabilizing CMFD," *Nuclear Science and Engineering*, Vol. 190, No. 1, 2018, pp. 45–55.

[19] TOTH, A. R., *A Theoretical Analysis of Anderson Acceleration and Its Application in Multiphysics Simulation for Light-Water Reactors*, Ph.D. thesis, North Carolina State University, 2016.

[20] Hamilton, S., Berrill, M., Clarno, K., Pawlowski, R., Toth, A., Kelley, C., Evans, T., and Philip, B., "An Assessment of Coupling Algorithms for Nuclear Reactor Core Physics Simulations," *Journal of Computational Physics*, Vol. 311, April 2016, pp. 241–257.

[21] Senecal, J. P. and Ji, W., "Approaches for Mitigating Over-Solving in Multiphysics Simulations," *International Journal for Numerical Methods in Engineering*, Vol. 112, No. 6, 2017, pp. 503–528.

[22] Senecal, J. P. and Ji, W., "Development of an Efficient Tightly Coupled Method for Multiphysics Reactor Transient Analysis," *Progress in Nuclear Energy*, Vol. 103, No. May 2017, 2018, pp. 33–44.

[23] Kelly, D. J., Kelly, A. E., Aviles, B. N., Godfrey, A. T., Salko, R. K., and Collins, B. S., "MC21/CTF and VERA Multiphysics Solutions to VERA Core Physics Benchmark Progression Problems 6 and 7," *Nuclear Engineering and Technology*, Vol. 49, No. 6, 2017, pp. 1326–1338.

[24] Liu, X. J. and Cheng, X., "Thermal-Hydraulic and Neutron-Physical Characteristics of a New SCWR Fuel Assembly," *Annals of Nuclear Energy*, Vol. 36, No. 1, 2009, pp. 28–36.

[25] Yu, J., Lee, H., Kim, H., Zhang, P., and Lee, D., "Preliminary Coupling of the Thermal/Hydraulic Solvers in the Monte Carlo Code MCS for Practical LWR Analysis," *Annals of Nuclear Energy*, Vol. 118, 2018, pp. 317–335.

[26] Gill, D. F., Griesheimer, D. P., and Aumiller, D. L., "Numerical Methods in Coupled Monte Carlo and Thermal-Hydraulic Calculations," *Nuclear Science and Engineering*, Vol. 185, No. 1, 2017, pp. 194–205.

[27] Daeubler, M., Ivanov, A., Sjenitzer, B. L., Sanchez, V., Stieglitz, R., and Macian-Juan, R., "High-Fidelity Coupled Monte Carlo Neutron Transport and Thermal-Hydraulic Simulations Using Serpent 2/SUBCHANFLOW," *Annals of Nuclear Energy*, Vol. 83, Sept. 2015, pp. 352–375.

[28] Yu, J., Lee, H., Kim, H., Zhang, P., and Lee, D., "Simulations of BEAVRS Benchmark Cycle 2 Depletion with MCS/CTF Coupling System," *Nuclear Engineering and Technology*, Vol. 52, No. 4, 2020, pp. 661–673.

[29] Chanaron, B., Ahnert, C., Crouzet, N., Sanchez, V., Kolev, N., Marchand, O., Kliem, S., and Papukchiev, A., "Advanced Multi-Physics Simulation for Reactor Safety in the Framework of the NURESAFE Project," *Annals of Nuclear Energy*, Vol. 84, Oct. 2015, pp. 166–177.

[30] Keyes, D. E., McInnes, L. C., Woodward, C., Gropp, W., Myra, E., Pernice, M., Bell, J., Brown, J., Clo, A., Connors, J., Constantinescu, E., Estep, D., Evans, K., Farhat, C., Hakim, A., Hammond, G., Hansen, G., Hill, J., Isaac, T., Jiao, X., Jordan, K., Kaushik, D., Kaxiras, E., Koniges, A., Lee, K., Lott, A., Lu, Q., Magerlein, J., Maxwell, R., McCourt, M., Mehl, M., Pawlowski, R., Randles, A. P., Reynolds, D., Rivière, B., Rüde, U., Scheibe, T., Shadid, J., Sheehan, B., Shephard, M., Siegel, A., Smith, B., Tang, X., Wilson, C., and Wohlmuth, B., "Multiphysics Simulations: Challenges and Opportunities," *International Journal of High Performance Computing Applications*, Vol. 27, No. 1, 2013, pp. 4–83.

[31] Toth, A. and Kelley, C. T., "Convergence Analysis for Anderson Acceleration," *SIAM Journal on Numerical Analysis*, Vol. 53, No. 2, Jan. 2015, pp. 805–819.

[32] Yee, B. C., Kochunas, B., and Larsen, E. W., "A Multilevel in Space and Energy Solver for 3-D Multigroup Diffusion and Coarse-Mesh Finite Difference Eigenvalue Problems," *Nuclear Science and Engineering*, Vol. 193, No. 7, July 2019, pp. 722–745.

[33] Liu, Z., Wang, B., Zhang, M., Zhou, X., and Cao, L., "An Internal Parallel Coupling Method Based on NECP-X and CTF and Analysis of the Impact of Thermal–Hydraulic Model to the High-Fidelity Calculations," *Annals of Nuclear Energy*, Vol. 146, 2020.

[34] Walker, E. D., Collins, B., and Gehin, J. C., "Low-Order Multiphysics Coupling Techniques for Nuclear Reactor Applications," *Annals of Nuclear Energy*, Vol. 132, Oct. 2019, pp. 327–338.

[35] Cattaneo, P., Lenain, R., Merle, E., Patricot, C., and Schneider, D., "Numerical Optimization of a Multiphysics Calculation Scheme Based on Partial Convergence," *Annals of Nuclear Energy*, Vol. 151, Feb. 2021, pp. 107892.

[36] Knoll, D. A. and Keyes, D. E., "Jacobian-Free Newton-Krylov Methods: A Survey of Approaches and Applications," *Journal of Computational Physics*, Vol. 193, No. 2, 2004, pp. 357–397.

[37] Anderson, D. G., "Iterative Procedures for Nonlinear Integral Equations," *Journal of the ACM*, Vol. 12, No. 4, Oct. 1965, pp. 547–560.

[38] Facchini, A., Lee, J., and Joo, H. G., "Investigation of Anderson Acceleration in Neutronics-Thermal Hydraulics Coupled Direct Whole Core Calculation," *Annals of Nuclear Energy*, Vol. 153, April 2021.

[39] Kochunas, B., Fitzgerald, A., and Larsen, E., "Fourier Analysis of Iteration Schemes for K-Eigenvalue Transport Problems with Flux-Dependent Cross Sections," *Journal of Computational Physics*, Vol. 345, Sept. 2017, pp. 294–307.

[40] Joo, H. G., "Transient Capability of the DeCART Code," *Transactions of American Nuclear Society*, American Nuclear Society, May 2004.

[41] Choi, S., Choe, J., Hoang, K., Nguyen, N., Lee, W., Kim, W., Kim, H., Ebiwonjumi, B., Jeong, E., Lee, H., Kim, K., Kim, K., and Lee, D., "Recent Development Status of Neutron Transport Code STREAM," *Transactions of the Korean Nuclear Society Spring Meeting*, Jeju, Korea, 2019.

[42] Zhu, A., Xu, Y., Graham, A., Young, M., Downar, T., and Cao, L., "Transient Methods for Pin-Resolved Whole Core Transport Using the 2D-1D Methodology in MPACT," *Proceedings of ANS MC2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*, Vol. 3, American Nuclear Society, Nashville, TN, 2015, pp. 2360–2375.

[43] Talamo, A., "Numerical Solution of the Time Dependent Neutron Transport Equation by the Method of the Characteristics," *Journal of Computational Physics*, Vol. 240, May 2013, pp. 248–267.

[44] Hoffman, A. J. and Lee, J. C., *A Time-Dependent Neutron Transport Method of Characteristics Formulation with Time Derivative Propagation*, Ph.D. thesis, University of Michigan, Ann Arbor, Feb. 2016.

[45] Ban, Y., Endo, T., and Yamamoto, A., "A Unified Approach for Numerical Calculation of Space-Dependent Kinetic Equation," *Journal of Nuclear Science and Technology*, Vol. 49, No. 5, May 2012, pp. 496–515.

[46] Hsieh, A., Zhang, G., and Yang, W. S., "Consistent Transport Transient Solvers of the High-Fidelity Transport Code PROTEUS-MOC," *Nuclear Science and Engineering*, Vol. 194, No. 7, July 2020, pp. 508–540.

[47] Tsujita, K., Endo, T., and Yamamoto, A., "Application of the Multigrid Amplitude Function Method for Time-Dependent MOC Based on the Linear Source Approximation," *Journal of Nuclear Science and Technology*, Vol. 57, American Nuclear Society, Sun Valley, ID, 2020, pp. 646–662.

[48] Kochunas, B., Downar, T., Liu, Y., Shen, Q., Dodson, Z., Gerlach, A., Jabaay, D., and Wysocki, A., "VERA-CS Transient Analysis for RIA," Tech. Rep. CASL-U-2017-1303-000, CASL, 2017.

[49] Kucukboyaci, V. N., Kochunas, B., Downar, T. J., Wysocki, A., and Salko, R. K., "Evaluation of VERA-CS Transient Capability for Analyzing the AP1000 Reactor Control Rod Ejection Accident," *Proceedings of PHYSOR 2018: Reactors Physics Paving the Way towards More Efficient Systems*, Cancun, Mexico, 2018.

[50] Shen, Q., Wang, Y., Jabaay, D., Kochunas, B., and Downar, T., "Transient Analysis of C5G7-TD Benchmark with MPACT," *Annals of Nuclear Energy*, Vol. 125, Jan. 2019, pp. 107–120.

[51] Zhu, A., *Transient Methods for Pin-Resolved Whole-Core Neutron Transport (Doctoral Dissertation)*, Ph.D. thesis, University of Michigan, 2016.

[52] Ott, K. O. and Meneley, D. A., "Accuracy of the Quasistatic Treatment of Spatial Reactor Kinetics," *Nuclear Science and Engineering*, Vol. 36, No. 3, June 1969, pp. 402–411.

[53] Dulla, S., Mund, E. H., and Ravetto, P., "Accuracy of a Predictor-Corrector Quasi-Static Method for Space-Time Reactor Dynamics," *Proceedings of PHYSOR 2006: American Nuclear Society's Topical Meeting on Reactor Physics*, American Nuclear Society, Vancouver, BC, 2006.

[54] Zhu, A., Xu, Y., and Downar, T., "A Multilevel Quasi-Static Kinetics Method for Pin-Resolved Transport Transient Reactor Analysis," *Nuclear Science and Engineering*, Vol. 182, No. 4, April 2016, pp. 435–451.

[55] Cao, L., Gerlach, A., Xu, Y., Downar, T., and Lee, J. C., "Neutronics Modeling of the SPERT III E-Core Critical Experiments with MPACT and KENO," *Annals of Nuclear Energy*, Vol. 80, June 2015, pp. 207–218.

[56] "REACTIVITY ACCIDENT TEST RESULTS AND ANALYSES FOR THE SPERT III E-CORE: A SMALL, OXIDE-FUELED, PRESSURIZED-WATER REACTOR." Tech. rep., United States, Jan. 1969.

[57] LeVeque, R. J., *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, Society for Industrial and Applied Mathematics, Jan. 2007.

[58] Leppänen, J., Pusa, M., Viitanen, T., Valtavirta, V., and Kaltiaisenaho, T., "The Serpent Monte Carlo Code: Status, Development and Applications in 2013," *Annals of Nuclear Energy*, Vol. 82, 2015, pp. 142–150.

[59] Wang, K., Li, Z., She, D., Liang, J., Xu, Q., Qiu, Y., Yu, J., Sun, J., Fan, X., and Yu, G., "RMC - A Monte Carlo Code for Reactor Core Analysis," *Annals of Nuclear Energy*, Vol. 82, Aug. 2015, pp. 121–129.

[60] Askew, J. R., "A Characteristics Formulation of the Neutron Transport Equation in Complicated Geometries," Tech. Rep. AEEW-M–1108, United Kingdom Atomic Energy Authority, 1972.

[61] Stimpson, S. G., *An Azimuthal, Fourier Moment-Based Axial SN Solver for the 2D/1D Scheme*, Ph.D. thesis, University of Michigan, 2015.

[62] Jarrett, M., Kochunas, B., Larsen, E., and Downar, T., "Polar Parity for Efficient Evaluation of Anisotropic Transverse Leakage in the 2D/1D Transport Method," *Nuclear Science and Engineering*, Vol. 193, No. 12, 2019, pp. 1291–1309.

[63] Kochunas, B. M., *A Hybrid Parallel Algorithm for the 3-D Method of Characteristics Solution of the Boltzmann Transport Equation on High Performance Compute Clusters*, Ph.D. thesis, University of Michigan, 2013.

[64] Cho, J. Y., Kim, K. S., Lee, C. C., Zee, S. Q., and Joo, H. G., "Axial SPN and Radial MOC Coupled Whole Core Transport Calculation," *Journal of Nuclear Science and Technology*, Vol. 44, No. 9, 2007, pp. 1156–1171.

[65] Team, M., *MPACT Theory Manual, Version 2.2.0*, 2020-Nov.

[66] Liu, Y., Martin, W., Williams, M., and Kim, K. S., "A Full-Core Resonance Self-Shielding Method Using a Continuous-Energy Quasi-One-Dimensional Slowing-down Solution That Accounts for Temperature-Dependent Fuel Subregions and Resonance Interference," *Nuclear Science and Engineering*, Vol. 180, No. 3, 2015, pp. 247–272.

[67] Duderstadt, J. J. and Hamilton, L. J., *Nuclear Reactor Analysis*, Wiley, New York, 1976.

[68] Kim, K. S., Gentry, C. A., Godfrey, A. T., Liu, Y., and Palmtag, S., "Development of the Multigroup Cross Section Library for the CASL Neutronics Simulator MPACT: Verification," *Annals of Nuclear Energy*, Vol. 132, 2019, pp. 1–23.

[69] Yee, B. C., Larsen, E. W., and Kochunas, B., "An Analytical Derivation of Transport-Corrected P0 Cross Sections and Diffusion Coefficients," *Proceedings of PHYSOR 2016: Unifying Theory and Experiments in the 21st Century*, American Nuclear Society, Sun Valley, ID, 2016.

[70] Lewis, E. E., "Fundamentals of Nuclear Reactor Physics," *Fundamentals of Nuclear Reactor Physics*, 2008.

[71] Ferrer, R. M. and Rhodes, J. D., "A Linear Source Approximation Scheme for the Method of Characteristics," *Nuclear Science and Engineering*, Vol. 182, No. 2, Feb. 2016, pp. 151–165.

[72] Fitzgerald, A., *Parallel 3-D Method of Characteristics with Linear Source and Advanced Transverse Integration*, PhD Thesis, University of Michigan, 2020.

[73] Liu, Y., Vaughn, K., Kochunas, B., and Downar, T., "Validation of Pin-Resolved Reaction Rates, Kinetics Parameters, and Linear Source MOC in MPACT," *Nuclear Science and Engineering*, Vol. 195, No. 1, Jan. 2021, pp. 50–68.

[74] Adams, M. L. and Larsen, E. W., "Fast Iterative Methods for Discrete-Ordinates Particle Transport Calculations," *Progress in Nuclear Energy*, Vol. 40, No. 1, Jan. 2002, pp. 3–159.

[75] Jarrett, M., Kochunas, B., Zhu, A., and Downar, T., "Analysis of Stabilization Techniques for CMFD Acceleration of Neutron Transport Problems," *Nuclear Science and Engineering*, Vol. 184, No. 2, Oct. 2016, pp. 208–227.

[76] Kochunas, B., "Theoretical Convergence Rate Analysis of a Unified CMFD Formulation with Various Diffusion Coefficients," *Proceedings of International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C 2019)*, American Nuclear Society, Portland, OR, 2019, pp. 978–990.

[77] Saad, Y., *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, 2nd ed., Jan. 2003.

[78] Yee, B. C., Kochunas, B., Larsen, E. W., and Xu, Y., "Space-Dependent Wielandt Shifts for Multigroup Diffusion Eigenvalue Problems," *Nuclear Science and Engineering*, Vol. 188, No. 2, Nov. 2017, pp. 140–159.

[79] Cornejo, L. R., Anistratov, D. Y., and Smith, K., "Iteration Methods with Multigrid in Energy for Eigenvalue Neutron Diffusion Problems," *Nuclear Science and Engineering*, Vol. 193, No. 8, 2019, pp. 803–827.

[80] Zhang, G., Hsieh, A., Yang, W. S., and Jung, Y. S., "Consistent pCMFD Acceleration Schemes of the Three-Dimensional Transport Code PROTEUS-MOC," *Nuclear Science and Engineering*, Vol. 193, No. 8, 2019, pp. 828–853.

[81] Briggs, W. L., Henson, V. E., and McCormick, S. F., *A Multigrid Tutorial*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2nd ed., 2000.

[82] Collins, B. and Stimpson, S., "Acceleration Methods for Whole Core Reactor Simulations Using VERA," *Transactions of the American Nuclear Society*, Vol. 118, Philadelphia, PA, 2018, pp. 929–932.

[83] Knoll, D. A., Park, H., and Newman, C., "Acceleration of k -Eigenvalue/Criticality Calculations Using the Jacobian-Free Newton-Krylov Method," *Nuclear Science and Engineering*, Vol. 167, No. 2, Feb. 2011, pp. 133–140.

[84] Aviles, B. N., Kelly, D. J., Aumiller, D. L., Gill, D. F., Siebert, B. W., Godfrey, A. T., Collins, B. S., and Salko, R. K., "MC21/COBRA-IE and VERA-CS Multiphysics Solutions to VERA Core Physics Benchmark Problem #6," *Progress in Nuclear Energy*, Vol. 101, 2017, pp. 338–351.

[85] Kochunas, B., "Demonstration of Neutronics Coupled to Thermal-Hydraulics for a Full-Core Problem Using COBRA-TF/MPACT," CASL Document CASL-U-2014-0051-000, 2014.

[86] Walker, E. D., *Low-Order Multiphysics Coupling Techniques for Nuclear Reactor Applications*, Ph.D. thesis, The University of Tennessee, 2017.

[87] Toth, A., Kelley, C. T., Slattery, S., Hamilton, S., Clarno, K., and Pawlowski, R., "Analysis of Anderson Acceleration on a Simplified Neutronics/Thermal Hydraulics System," *Proceedings of ANS MC2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*, Vol. 4, American Nuclear Society, Nashville, TN, 2015, pp. 2589–2600.

[88] Ott, K. O. and Neuhold, R. J., *Introductory Nuclear Reactor Dynamics, American Nuclear Society*, American Nuclear Society, 1985.

[89] Graham, A., Downar, T., Collins, B., Salko, R., and Palmtag, S., "Assessment of Thermal-Hydraulic Feedback Models," *Proceedings of PHYSOR 2016: Unifying Theory and Experiments in the 21st Century*, Vol. 6, American Nuclear Society, Sun Valley, ID, 2016, pp. 3616–3626.

[90] Salko, R. K. and Avramova, M. N., "CTF Theory Manual," Tech. rep., The North Carolina State University.

[91] Heath, M. T., *Scientific Computing: An Introductory Survey*, No. 80 in Classics in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, revised second edition, siam edition ed., 2018.

[92] Yang, W. S., Downar, T., Hsieh, A., Jabaay, D., Li, J., Wharry, J. P., and Lee, C., "Development of Transient Capabilities for the NEAMS Neutronics Code PROTEUS," Tech. Rep. DOE-PUR-0008561, 1581446, Dec. 2019.

[93] Kochunas, B., Collins, B., Jabaay, D., Downar, T. J., and Martin, W. R., "Overview of Development and Design of MPACT: Michigan Parallel Characteristics Transport Code," *Proceedings of International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C 2013)*, Vol. 1, Sun Valley, ID, 2013, pp. 42–53.

[94] Zhu, A., Kochunas, B., Xu, Y., Jarrett, M., Larsen, E., and Downar, T., "Theoretical Convergence Rate Lower Bounds for Variants of Coarse Mesh Finite Difference to Accelerate Neutron Transport Calculations," *Nuclear Science and Engineering*, Vol. 186, No. 3, June 2017, pp. 224–238.

[95] Shen, Q., Xu, Y., and Downar, T., "Stability Analysis of the CMFD Scheme with Linear Prolongation," *Annals of Nuclear Energy*, Vol. 129, July 2019, pp. 298–307.

[96] Saad, Y. and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869.

[97] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W. D., Karpeyev, D., Kaushik, D., Knepley, M. G., Curfman McInnes, L., Rupp, K., Smith, B. F., Zampini, S., and Zhang, H., "PETSc Users Manual," Tech. Rep. ANL-95/11 - Revision 3.13, Argonne National Laboratory, 2015.

[98] Godfrey, A. T., "VERA Core Physics Benchmark Progression Problem Specifications," Tech. Rep. 793, CASL, 2014.

[99] Herman, B. R., *Monte Carlo and Thermal Hydraulic Coupling Using Low-Order Nonlinear Diffusion Acceleration by Certified By*, Ph.D. thesis, MIT, 2014.

[100] Wang, J., Wang, Q., and Ding, M., "Review on Neutronic/Thermal-Hydraulic Coupling Simulation Methods for Nuclear Reactor Analysis," *Annals of Nuclear Energy*, Vol. 137, March 2020, pp. 107165.

[101] Herman, B. R., Forget, B., and Smith, K., "Progress toward Monte Carlo–Thermal Hydraulic Coupling Using Low-Order Nonlinear Diffusion Acceleration Methods," *Annals of Nuclear Energy*, Vol. 84, Oct. 2015, pp. 63–72.

[102] Zhong, Z., Downar, T. J., Xu, Y., DeHart, M. D., and Clarno, K. T., "Implementation of Two-Level Coarse-Mesh Finite Difference Acceleration in an Arbitrary Geometry, Two-Dimensional Discrete Ordinates Transport Method," *Nuclear Science and Engineering*, Vol. 158, No. 3, 2008, pp. 289–298.

[103] Adams, B. T. and Morel, J. E., "Two-Grid Acceleration Scheme for the Multigroup Sn Equations with Neutron Upscattering," *Airport Pavement Innovations Theory to Practice*, Vol. 115, No. 3, 1993, pp. 253–264.

[104] Wysocki, A., Gerlach, A., Kochunas, B., and Salko, R., "Initial Implementation of Transient VERA-CS," Tech. Rep. CASL-U-2017-1303-000, CASL, 2017.

[105] Kochunas, B., Zhu, A., Jabaay, D., Xu, Y., and Downar, T., "Whole-Core Pin-Resolved PWR Transient Calculations in MPACT," *Proceedings of International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C 2017)*, Jeju, Korea, 2017.

[106] Dutt, A., Greengard, L., and Rokhlin, V., "Spectral Deferred Correction Methods for Ordinary Differential Equations," *BIT Numerical Mathematics*, Vol. 40, No. 2, 2000, pp. 241–266.

[107] Beylkin, D., "Spectral Deferred Corrections for Parabolic Partial Differential Equations," Tech. rep., Yale University, New Haven, CT, 2015.

[108] Böhmer, K. and Stetter, H., *Defect Correction Methods*, Vol. 5 of *Computing Supplementum*, Springer Vienna, Vienna, Nov. 1984.

[109] Cai, Y., Peng, X., Li, Q., Zhang, Z., Jiang, Z., and Rui, G., "Parallel Computation of the Point Neutron Kinetic Equations Using Parallel Revisionist Integral Deferred Correction," *Proceedings of 2018 26th International Conference on Nuclear Engineering*, American Society of Mechanical Engineers, London, England, July 2018, p. V003T02A010.

[110] Cai, Y., Li, Q., and Wang, K., "A Numerical Solution to the Point Kinetic Equations Using Spectral Deferred Correction," *Transactions of the American Nuclear Society*, Vol. 111, Anaheim, 2014, pp. 734–736.

[111] Minion, M. L., "Semi-Implicit Spectral Deferred Correction Methods for Ordinary Differential Equations," *Communications in Mathematical Sciences*, Vol. 1, No. 3, 2003, pp. 471–500.

[112] Crockatt, M. M., Christlieb, A. J., Garrett, C. K., and Hauck, C. D., "An Arbitrary-Order, Fully Implicit, Hybrid Kinetic Solver for Linear Radiative Transport Using Integral Deferred Correction," *Journal of Computational Physics*, Vol. 346, Oct. 2017, pp. 212–241.

[113] Huang, D. Z., Pazner, W., Persson, P. O., and Zahr, M. J., "High-Order Partitioned Spectral Deferred Correction Solvers for Multiphysics Problems," *Journal of Computational Physics*, Vol. 412, 2020, pp. 109441.

[114] Greengard, L., "Spectral Integration and Two-Point Boundary Value Problems," *SIAM Journal on Numerical Analysis*, Vol. 28, No. 4, 1991, pp. 1071–1080.

[115] Causley, M. F. and Seal, D. C., "On the Convergence of Spectral Deferred Correction Methods," *Communications in Applied Mathematics and Computational Science*, Vol. 14, No. 1, 2019, pp. 33–64.

[116] Hagstrom, T. and Zhou, R., "On the Spectral Deferred Correction of Splitting Methods for Initial Value Problems," *Communications in Applied Mathematics and Computational Science*, Vol. 1, No. 1, Dec. 2006, pp. 169–205.

[117] Huang, J., Jia, J., and Minion, M., "Accelerating the Convergence of Spectral Deferred Correction Methods," *Journal of Computational Physics*, Vol. 214, No. 2, May 2006, pp. 633–656.

[118] Boffie, J. and Pounders, J., "An Adaptive Time Step Control Scheme for the Transient Diffusion Equation," *Annals of Nuclear Energy*, Vol. 116, June 2018, pp. 280–289.

[119] Hackemack, M. W. and Pounders, J. M., "Implementation of an a Priori Time Step Estimator for the Multigroup Neutron Diffusion Equation in Asynchronously Coupled Relap5-3d," *Proceedings of PHYSOR 2014: The Role of Reactor Physics toward a Sustainable Future*, Kyoto, Japan, 2014.

[120] Gerlach, A., *Adaptive Time Stepping for the Neutron Transport Solution with the Alpha Eigenvalue*, Ph.D. thesis, University of Michigan, 2020.

[121] Söderlind, G., "Digital Filters in Adaptive Time-Stepping," *ACM Transactions on Mathematical Software*, Vol. 29, No. 1, March 2003, pp. 1–26.

[122] Buckner, M. R. and Stewart, J. W., "Multidimensional Space-Time Nuclear-Reactor Kinetics Studies—Part I: Theoretical," *Nuclear Science and Engineering*, Vol. 59, No. 4, April 1976, pp. 289–297.

[123] Ferguson, D. R. and Hansen, K. F., "Solution of the Space-Dependent Reactor Kinetics Equations in Three Dimensions," *Nuclear Science and Engineering*, Vol. 51, No. 2, June 1973, pp. 189–205.

[124] Reed, W. H. and Hansen, K. F., "Alternating Direction Methods for the Reactor Kinetics Equations," *Nuclear Science and Engineering*, Vol. 41, No. 3, Sept. 1970, pp. 431–442.

[125] Stimpson, S., Collins, B., and Downar, T., "A 2-D/1-D Transverse Leakage Approximation Based on Azimuthal, Fourier Moments," *Nuclear Science and Engineering*, Vol. 185, No. 2, Feb. 2017, pp. 243–262.

[126] Nguyen, T. N., Jung, Y. S., Downar, T., and Lee, C., "Implementation of the Transient Fixed-Source Problem in the Neutron Transport Code PROTEUS-MOC," *Annals of Nuclear Energy*, Vol. 129, July 2019, pp. 199–206.

[127] Caron, D., Dulla, S., and Ravetto, P., "Adaptive Time Step Selection in the Quasi-Static Methods of Nuclear Reactor Dynamics," *Annals of Nuclear Energy*, Vol. 105, July 2017, pp. 266–281.

[128] *Solving Ordinary Differential Equations I*, Vol. 8 of *Springer Series in Computational Mathematics*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.

[129] Zhu, A., Xu, Y., and Downar, T., "Stability Analysis of the Backward Euler Time Discretization for the Pin-Resolved Transport Transient Reactor Calculation," *Annals of Nuclear Energy*, Vol. 87, Jan. 2016, pp. 252–266.