

TEORIA DE DIFUSÃO DE 1 GRUPO MODIFICADO APLICADA À ACELERAÇÃO DOS CÁLCULOS DE OTIMIZAÇÃO DE NÚCLEO

Marco Antonio Chaves Lima

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Nuclear, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Nuclear.

Orientadores: Aquilino Senra Martinez
Alessandro da Cruz Gonçalves
Adilson Costa da Silva

Rio de Janeiro
Março de 2026

TEORIA DE DIFUSÃO DE 1 GRUPO MODIFICADO APLICADA À
ACELERAÇÃO DOS CÁLCULOS DE OTIMIZAÇÃO DE NÚCLEO

Marco Antonio Chaves Lima

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA
DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR
EM CIÊNCIAS EM ENGENHARIA NUCLEAR.

Orientadores: Aquilino Senra Martinez
Alessandro da Cruz Gonçalves
Adilson Costa da Silva

Aprovada por: Prof. Aquilino Senra Martinez
Prof. Alessandro da Cruz Gonçalves
Prof. Adilson Costa da Silva
Prof. Hermes Alves Filho
Prof. Sérgio de Oliveira Vellozo
Prof. Antônio Carlos de Abreu Mol

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2026

Lima, Marco Antonio Chaves

Teoria de Difusão de 1 Grupo Modificado Aplicada à Aceleração dos Cálculos de Otimização de Núcleo/Marco Antonio Chaves Lima. – Rio de Janeiro: UFRJ/COPPE, 2026.

XIV, 88 p.: il.; 29,7cm.

Orientadores: Aquilino Senra Martinez

Alessandro da Cruz Gonçalves

Adilson Costa da Silva

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Nuclear, 2026.

Referências Bibliográficas: p. 72 – 74.

1. Teoria da Difusão de 1 Grupo Modificado. 2. Redução do Tempo Computacional. 3. Otimizações do Compilador. I. Martinez, Aquilino Senra *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Nuclear. III. Título.

Dedico esta tese à minha mãe.

Agradecimentos

A Deus, por sempre segurar nas minha mãos, por nunca me abandonar, por nunca deixar de me escutar e de me acalmar, por estar ao meu lado nos momentos calmos e nos turbulentos. Sem Deus não teria sido possível chegar até aqui.

À minha mãe, Creusa, pelo amor, compreensão, apoio incondicional e por sempre interceder por mim em suas orações.

Aos meus irmãos, Andréia, Déborah e Marcelo pelo apoio incondicional preocupação e pelo carinho de sempre.

Ao professor Aquilino Senra Martinez, pela receptividade, paciência, confiança, experiência e pelos ensinamentos ao longo deste trabalho.

Ao professor Adilson Costa da Silva, pela orientação, paciência, apoio pedagógico e emocional, e por me manter calmo nos momentos mais tensos.

Ao professor Alessandro da Cruz Gonçalves, pela receptividade, incentivo e sugestões que fizeram uma enorme diferença no desenvolvimento desta tese.

Ao professor Fernando Carvalho da Silva, por ter gentilmente cedido a programação do benchmark de queima.

Ao Programa de Engenharia Nuclear (PEN/COPPE/UFRJ), pelo suporte e apoio institucional.

Ao Conselho Nacional de Desenvolvimento e Pesquisa (CNPq), pelo financiamento através de bolsa de estudos, que sem ele não teria sido possível chegar aqui.

Aos professores que gentilmente se disponibilizaram a participar de minha banca.

Aos amigos do laboratório, William Abreu, Artur Sousa, Lionel Sallas, Alefe Ribeiro, Leonardo Kenji, Júlia Pontes e Flávio Lopes, por formarem um grupo tão divertido e um ambiente tão leve, descontraído e bom de conviver. Adoro vocês!!!

Aos meus amigos Edimar Ferreira da Silva e André Martins, pela amizade e momentos de descontração.

Ao meu amigo Umberto Cassará, que me incentivou a entrar na pós graduação.

À minha amiga Mônica, pelo cuidado, amizade e momentos de descontração.

À minha amiga Juliana Pereira, que é minha amiga desde a graduação e agora é minha colega de Doutorado.

In memoriam, à servidora Sheila de Matos Imamura, diretora da Residência Estudantil da UFRJ, por tudo o que fez por mim em vida. Do fundo do meu

coração, ficarei para sempre agradecido!

À todas as pessoas aqui não mencionadas, mas que contribuíram direta ou indiretamente para este trabalho e pela minha jornada até aqui.

Agradecimento Especial à UFRJ

A finalização desta tese marca o fim do meu período como estudante nesta instituição. Que com todos os defeitos e críticas que eu possa fazer, a UFRJ foi, é, e será sempre, um lugar que ficará no meu coração. De todas as pessoas maravilhosas conheci; de todos os grandes professores que tive o prazer de assistir aula; dos técnicos, servidores, funcionários terceirizados e trabalhadores autônomos que fazem a roda da universidade girar, e com muitos dos quais contei com ajuda nos momentos que mais precisei, e sempre tiveram um carinho enorme comigo.

A UFRJ foi, é, e continuará sendo a porta de entrada para muitos estudantes, que como eu, não teriam uma outra oportunidade se não fosse ela. Que essa porta permaneça aberta, para que muitos realizem seus sonhos de ter acesso à educação superior.

Fica aqui minha consideração, carinho e respeito à esta instituição onde passei os melhores (e piores) momentos da minha vida, e que nunca deixará de fazer parte de mim! Aonde eu for, a UFRJ irá comigo.

Obrigado.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

TEORIA DE DIFUSÃO DE 1 GRUPO MODIFICADO APLICADA À ACELERAÇÃO DOS CÁLCULOS DE OTIMIZAÇÃO DE NÚCLEO

Marco Antonio Chaves Lima

Março/2026

Orientadores: Aquilino Senra Martinez

Alessandro da Cruz Gonçalves

Adilson Costa da Silva

Programa: Engenharia Nuclear

Um reator nuclear de potência opera em ciclos que duram por um certo período com uma determinada composição material. Ao final desse período, o reator é desligado e reabastecido para o próximo ciclo. Essa nova configuração deve atender a critérios econômicos, de segurança e operacionais. Essa tarefa implica na simulação de inúmeras configurações candidatas e requer um código neutrônico que resolva a equação de difusão de nêutrons em multigrupos de energia, para que se possa determinar as características de cada configuração. Devido ao elevado custo computacional, aproximações do modelo de difusão são necessárias para reduzir o tempo de processamento dessa tarefa. O objetivo desta tese é utilizar a teoria de difusão de um grupo modificado como ferramenta de redução do tempo computacional na otimização do núcleo. Para acelerar ainda mais os cálculos de difusão, otimizações do compilador também foram feitas sem a necessidade de programação paralela ou alterações no código. Para a validação da teoria de um grupo modificado foram adotados resultados de referência disponíveis na literatura. E para os casos envolvendo diferentes configurações de núcleos, a teoria da difusão com dois grupos de energia foi adotada como referência a fim de avaliar a acurácia da teoria de um grupo modificado, bem como a redução do tempo computacional. As simulações mostraram que o uso da teoria da difusão com 1 grupo modificado sem otimizações no compilador reduzem o tempo computacional em 50%, e com otimizações, cerca de 80%, com razoável precisão nos resultados dos cálculos neutrônicos.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

MODIFIED ONE-GROUP DIFFUSION THEORY APPLIED TO ACCELERATING CORE OPTIMIZATION CALCULATIONS

Marco Antonio Chaves Lima

March/2026

Advisors: Aquilino Senra Martinez
Alessandro da Cruz Gonçalves
Adilson Costa da Silva

Department: Nuclear Engineering

The nuclear power reactor operates in cycles that last for a certain period of time with a specific material composition. At the end of this period, the reactor is shut down and refueled for the next cycle. This new configuration must meet economic, safety, and operational criteria. This task involves simulating a large number of candidate configurations and requires a neutronic code that solves the neutron diffusion equation in multi-energy groups, so that the characteristics of each configuration can be determined. Due to the high computational cost, approximations of the diffusion model are needed to reduce the processing time of this task. The objective of this thesis is to use the modified one-group diffusion theory as a tool to reduce computational time in core optimization. To further accelerate diffusion calculations, compiler optimizations were also performed without the need for parallel programming or code changes. To validate the modified one-group diffusion theory, reference results available in literature were adopted. And for the case involving different core configurations, the two-group energy theory was adopted as reference to evaluate the accuracy of modified one-group diffusion theory, as well as the reducing of computational time. The simulation results showed that the using modified one-group diffusion theory without compiler optimizations reduces computational time by 50%, and with optimizations, around 80%, with reasonable accuracy of the neutronic calculations results.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xiii
1 Introdução	1
2 Equação da Difusão de Nêutrons	7
2.1 Introdução	7
2.2 Considerações Iniciais	7
2.3 Equação da Difusão para 2 Grupos de Energia	8
2.4 Equação da Difusão para 1 Grupo Modificado	9
3 Otimizações no Compilador	11
3.1 Introdução	11
3.2 Paralelização	11
3.3 Otimizações no Compilador Intel	12
3.3.1 Opções de Otimização	13
3.4 OpenMP	21
3.4.1 Introdução	21
3.4.2 Compartilhamento de Memória	22
3.4.3 Threads	24
4 Apresentação dos Benchmarks e do Gerador de Núcleos	29
4.1 Benchmark Biblis	30
4.2 Benchmark Koebke	31
4.3 Benchmark de Pesquisa de Criticalidade, Queima e Depleção	32
4.4 Gerador de Núcleos	37
5 Resultados	41
5.1 Biblis	42
5.2 Koebke	46
5.3 Análise de 1 ciclo completo	49

5.3.1	Benchmark 19	49
5.3.2	Resultados para outras configurações	54
6	Discussões e Conclusões	68
	Referências Bibliográficas	72
A	Equações de Depleção	75
B	Pesquisa de Criticalidade	78
C	Opções de Otimização	79
D	Queima Média Acumulada	87

Lista de Figuras

1.1	Vista superior do núcleo de um PWR.	2
1.2	Representação de 1/8 de núcleo da Figura 1.1	3
1.3	Representação de 1/4 de núcleo da Figura 1.1	4
3.1	Esquema do Compilador.	12
3.2	Estágios da Compilação.	13
3.3	Esquema da vetorização automática.	15
3.4	Link-Time Optimization - LTO	18
3.5	Representação Intermediária - IR.	19
3.6	Esquema de Memória Compartilhada Centralizada.	23
3.7	Esquema de Memória Distribuída.	23
3.8	Esquema do NUMA.	24
3.9	Esquema do COMA.	25
3.10	Implementação de Multithreading.	25
3.11	Modelo de Execução Fork-Join.	26
4.1	Núcleo do benchmark Biblis com 1/4 de simetria.	30
4.2	Núcleo do benchmark Koebke com 1/4 de simetria.	32
4.3	Núcleo do Benchmark 19 para o 1 ^o ciclo de carregamento com 1/8 de simetria.	33
4.4	Cadeia de depleção de núclídeos.	34
4.5	Esquema de codificação em chaves aleatórias de um vetor com 31 dimensões.	38
4.6	Núcleo resultante da decodificação em chaves aleatórias.	39
4.7	Núcleo resultante com os tipos em novas posições.	39
4.8	Núcleo resultante final incluindo tipos simétricos e tipo 8.	40
5.1	Valor do k_{eff} para 100 configurações do Biblis.	44
5.2	Desvio do k_{eff} para 100 configurações do Biblis.	44
5.3	Tempo médio de processamento para 100 configurações do Biblis.	45
5.4	Valor do k_{eff} para 100 configurações do Koebke.	47
5.5	Desvio do k_{eff} para 100 configurações do Koebke.	48

5.6	Tempo médio de processamento para 100 configurações do Koebke. . .	48
5.7	Curva de Boro.	52
5.8	Curva de Boro do Núcleo 93.	62
5.9	Curva de Boro do Núcleo 70.	62
5.10	Concentração de Boro - 0 DEPP.	64
5.11	Concentração de Boro - 30 DEPP.	64
5.12	Concentração de Boro - 150 DEPP.	65
5.13	Concentração de Boro - 270 DEPP.	65
5.14	Concentração de Boro - 381,49 DEPP.	66
5.15	Tempo médio de processamento para 100 configurações de 1 ciclo completo.	67
A.1	Cadeia usada para equações de depleção.	75
C.1	Geral.	79
C.2	Multiprocessadores.	80
C.3	Nível de Otimização.	80
C.4	Função Inline.	81
C.5	Tamanho ou Velocidade do Código.	81
C.6	Paralelização Automática.	82
C.7	Pré-carregamento de Cache.	82
C.8	Otimização Interprocedural.	83
C.9	Pré-processamento.	83
C.10	Geração do código com base nas extensões SIMD.	84
C.11	Geração do código para arquitetura específica.	84
C.12	Formato do Arquivo.	85
C.13	Diretivas para OpenMP.	85
C.14	Linha de comando.	86

Lista de Tabelas

4.1	Parâmetros nucleares - Biblis.	31
4.2	Parâmetros nucleares - Koebke.	32
4.3	Concentração crítica de boro $\times$ queima média por DEPP.	35
4.4	Densidades iniciais dos nuclídeos ($10^{24} \text{ átomos/cm}^3$).	35
4.5	Seções de choque microscópicas dos nuclídeos fissionáveis (10^{-24} cm^2).	36
4.6	Seções de choque microscópicas de captura do veneno queimável, xenônio, produtos de fissão e do boro solúvel.	36
4.7	Rendimentos do xenônio e dos produtos de fissão.	37
4.8	Seções de choque macroscópicas do material estrutural.	37
5.1	Resultados para o tempo de processamento do Biblis.	42
5.2	Distribuição de potência normalizada do Biblis.	43
5.3	Resultados para o tempo de processamento do Koebke.	46
5.4	Distribuição de potência normalizada do Koebke.	47
5.5	Distribuição de potência normalizada em BOC1 sem xenônio.	49
5.6	Distribuição de potência normalizada em EOC1.	50
5.7	Queima Média por EC em EOC1.	51
5.8	Concentração crítica de boro $\times$ queima média por DEPP.	52
5.9	Concentrações Médias em EOC1.	53
5.10	Resultados para o tempo de processamento.	53
5.11	Distribuição de potência normalizada em BOC1 antes do equilíbrio do xenônio do Núcleo 93.	55
5.12	Distribuição de potência normalizada em BOC1 antes do equilíbrio do xenônio do Núcleo 70.	56
5.13	Distribuição de potência normalizada em EOC1 do Núcleo 93.	57
5.14	Distribuição de potência normalizada em EOC1 do Núcleo 70.	58
5.15	Queima Média por EC em EOC1 do Núcleo 93.	59
5.16	Queima Média por EC em EOC1 do Núcleo 70.	60
5.17	Concentração crítica de boro $\times$ queima média do Núcleo 93.	61
5.18	Concentração crítica de boro $\times$ queima média do Núcleo 70.	61
5.19	Concentrações Médias em EOC1 do Núcleo 93.	63

5.20	Concentrações Médias em EOC1 do Núcleo 70.	63
5.21	Resultados para o tempo de processamento do Núcleo 93.	63
5.22	Resultados para o tempo de processamento do Núcleo 70.	63

Capítulo 1

Introdução

Um reator nuclear de potência é projetado para operar durante um certo período com uma determinada composição material. A este período chamamos de ciclo. Ao final deste ciclo, o nível de reatividade dos elementos combustíveis é reduzido, impossibilitando a operação do reator sob a condição de criticalidade. Então, seu desligamento torna-se necessário, terminando o ciclo, e uma nova composição do núcleo precisa ser definida para garantir a operação do reator pelo maior tempo possível.

Durante este processo, os elementos combustíveis (ECs) são levados à Piscina de Combustível Queimado¹ e cerca de 1/3 do combustível é substituído por elementos frescos. Os outros 2/3 do combustível são compostos por aqueles elementos que foram para a piscina e retornarão para serem reaproveitados e reposicionados na configuração do novo ciclo.

Da prática, sabe-se que cerca de 1/3 dos elementos combustíveis são substituídos a cada ciclo, gerando um acoplamento entre os ciclos, uma vez que a recarga parcial garante que uma fração do núcleo seja proveniente do ciclo anterior e outra que estará presente nos ciclos subsequentes. Essa combinação de elementos frescos e irradiados também tem efeito na reatividade do sistema, pois no plano de recarga é possível observar um aumento gradual do enriquecimento dos elementos frescos carregados no núcleo ao longo dos ciclos de operação². A predição da reatividade está relacionado ao limite de queima máxima que cada elemento pode atingir durante a operação, assim como o limite de queima máxima que cada vareta de combustível pode alcançar.

Os elementos combustíveis são posicionados no núcleo de forma simétrica, tal que, são classificados em razão da sua posição. A Figura 1.1 mostra o esquema do

¹Piscina de elementos combustíveis localizada dentro do prédio de contenção do reator nuclear, que é utilizada para armazenar temporariamente os combustíveis irradiados de forma segura.

²O enriquecimento inicial dos reatores PWR varia de 2 a 3%, podendo chegar ao limite máximo de enriquecimento para os reatores PWR comerciais licenciados, que é de 5%

núcleo de um reator PWR, destacando as linhas de simetria que dividem o núcleo em partes iguais e a classificação dos elementos quanto à posição (OLIVEIRA, 2013) . Por consequência dessa disposição há uma distribuição de potência simétrica no núcleo. E essa divisão em quadrantes ou octantes não apenas é útil na simulação de núcleo, mas será de muita utilidade como veremos mais adiante na reconfiguração do novo núcleo.

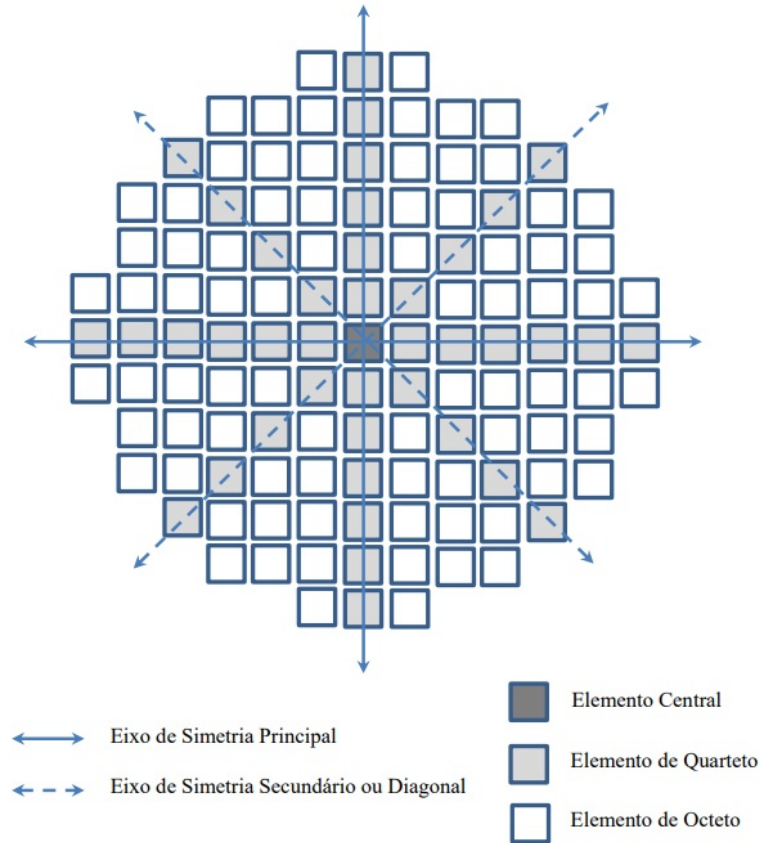


Figura 1.1: Vista superior do núcleo de um PWR.

Uma vez selecionados quais elementos combustíveis serão substituídos e quais serão reposicionados, o problema que se põe é como combinar esses diferentes ECs na nova configuração de tal forma que possibilite a continuação da produção de energia com maior duração do ciclo possível dentro dos limites de segurança especificados no projeto do reator. Dá-se a este problema o nome de *Problema de Otimização de Recarga* (POR), que, na verdade, trata-se de um problema combinatório que envolve a consideração de restrições e múltiplos objetivos (por vezes, conflitantes entre si) relacionados com aspectos econômicos, de segurança e operacionais (GOLDBARG, 2005) .

O problema de otimização de recarga consiste numa análise detalhada de muitas configurações de forma a avaliar se atendem os critérios de segurança, operacionais, respeitando-se o tipo de carregamento requerido pelo operador, seja posicionado

os elementos frescos na periferia do núcleo, ou em seu interior. Por isso, as configurações candidatas são geradas permutando-se seus n elementos, totalizando $n!$ configurações possíveis³. No entanto, o tempo para avaliar todas essas configurações candidatas é impraticável. Como exemplo, se considerássemos a substituição de todos os elementos do reator da Central Nuclear de Angra 1 (CNA1), com 121 elementos, seria necessário avaliar 121! permutações, ou aproximadamente 8.09×10^{200} configurações distintas.

Como forma de contornar este problema, modelagens do núcleo são feitas levando-se em conta sua simetria, reduzindo de forma significativa o número de configurações possíveis. Por isso, geralmente são feitas modelagens que representam fatias de 1/4 (quarteto) ou 1/8 (octeto) do núcleo original, como mostram as Figuras 1.2 e 1.3, referentes à Figura 1.1. Ainda sim, o número de permutações ainda continua grande, isto é, a simplificação do problema, não necessariamente, diminui sua complexidade.

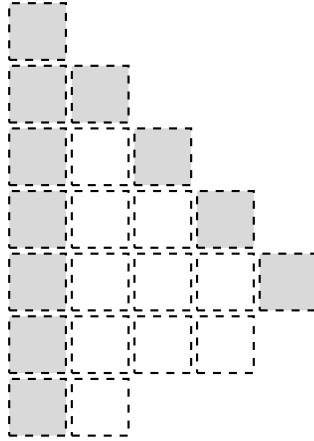


Figura 1.2: Representação de 1/8 de núcleo da Figura 1.1

Para além do problema combinatório, uma configuração candidata a de recarga só pode ser assim chamada quando avaliadas as características neutrônicas de seu núcleo. Essa avaliação consiste em cálculos neutrônicos para cada configuração. E para isso, é necessário resolver a equação da difusão de nêutrons em multigrupos de energia para conhecermos como estão distribuídos os nêutrons dentro do núcleo, ou seja, como é o fluxo de nêutrons no núcleo, e, a partir dele, calcular a distribuição de potência, queima, depleção e a estimar o comprimento do ciclo da configuração candidata à recarga. E daí, poder-se-ia dizer ou não se uma dada configuração atende aos critérios de segurança e operacionais. A resolução da equação da difusão de nêutrons é feita através da discretização espacial, dividindo o núcleo do reator

³É importante salientar que não necessariamente todas essas configurações são candidatas, pois é óbvio que nem todas atendem aos critérios de segurança e operacionais.

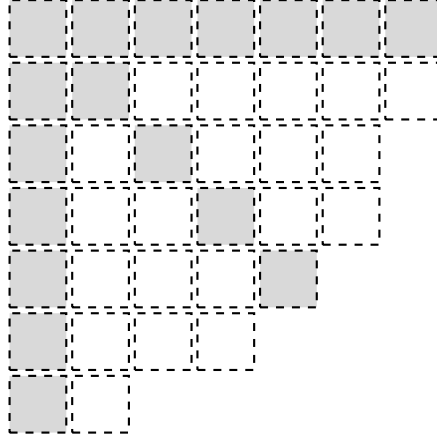


Figura 1.3: Representação de 1/4 de núcleo da Figura 1.1

em milhares de regiões de cálculo com alto custo de processamento, pois a solução em milhares pontos de malha constitui-se um enorme problema computacional. E sabendo-se que para qualquer processo de otimização, o custo computacional está diretamente ligado ao desempenho dos códigos de física de reatores, torna-se impraticável a avaliação de todas as possíveis configurações candidatas à nova configuração.

Diante deste desafio, a escolha da nova configuração do núcleo, respeitando as restrições impostas no início do problema, deve envolver técnicas computacionais diversas e cada vez mais eficientes, de forma a encontrar uma resposta num tempo razoável para o padrão de carregamento (PC) do núcleo. Desde os anos de 1990, ocorreu um aumento significativo de publicações científicas na área de otimização de recarga de reatores nucleares. E o concomitante melhoramento das técnicas computacionais resultantes do avanço tecnológico dos computadores tem incentivado pesquisadores a buscar novas formas de otimização de recarga (OLIVEIRA, 2013). O aparecimento das técnicas de computação avançadas resultou na ampla utilização de **Algoritmos de Otimização de Recarga**, e, mais recentemente, em metaheurísticas baseadas em populações aplicadas à solução de problemas complexos (LI *et al.*, 2022). A otimização de um determinado problema é a busca por uma melhor solução dentro de um conjunto finito ou infinito de possíveis soluções para este problema. O que diferencia cada técnica de otimização é como a busca é feita. E assim, as várias técnicas de otimização dão origens a diferentes algoritmos. A escolha depende de uma série de características do problema, principalmente se houver uma função que o represente. No caso do POR, sua solução é muito complexa, visto que é um problema altamente não linear, representado por um conjunto de funções e alto custo computacional envolvido no processo de escolha do novo núcleo.

Mesmo com o avanço tecnológico dos computadores, como dito anteriormente, ainda são grandes consumidores de tempo computacional os códigos de física que são acoplados a qualquer técnica de otimização. Esses códigos, normalmente feitos em

malha grossa, em 3 dimensões e com dois grupos de energia, servem para gerar os dois parâmetros nucleares usados pelos algoritmos de otimização de recarga: Máxima Concentração de Boro (C_B) e o Fator de Pico de Potência Radial F_{XYmax} . Por essa razão, quanto mais eficientes e rápidos forem os códigos neutrônicos, aliados à técnicas de otimização, mais capazes serão de ampliar o espaço de busca de soluções durante o processo de otimização, permitindo um maior número de avaliações.

Pensando nisso, esta tese tem o objetivo de otimizar um código neutrônico para redução do tempo computacional dos cálculos de recarga. Para isso, utilizaremos a teoria da difusão de 1 grupo modificado aliado à otimizações no compilador, que, juntas, reduzem significativamente o tempo de processamento com razoável precisão. Essa redução significativa dos tempos de processamento computacional já foi demonstrada num artigo já publicado por DA SILVA *et al.* (2023), que faz parte do desenvolvimento desta tese. Os resultados para o reator Biblis mostraram que o uso da teoria da difusão de 1 grupo modificado juntamente com otimizações do compilador, reduziu os tempos computacionais do código usado para os cálculos do fator de multiplicação efetivo e da distribuição de potência. Os resultados mostraram, que os tempos de processamento com a teoria de 1 grupo modificado reduziram em mais de 50% quando comparados aos cálculos com a teoria da difusão de dois grupos de energia. E nas simulações com otimizações no compilador aplicadas à teoria de 1 grupo modificado, os tempos de processamento reduziram em mais de 80%, com desvios máximos de 5% e sem alterar nenhuma linha de código da programação e nenhum dos resultados. Embora comparassem apenas o fator de multiplicação efetivo e a distribuição de potência, a ideia de usar a teoria de 1 grupo modificado com otimizações do compilador em cálculos de recarga foi bem-sucedida, não apenas em relação à redução do tempo de processamento computacional, mas também em relação à precisão. Nesta tese, simulamos um benchmark considerando um ciclo completo de queima, fazendo pesquisa de criticalidade, queima e depleção isotópica, usando a teoria da difusão com dois grupos de energia e um grupo modificado. Também simulamos esse mesmo benchmark usando a teoria de 1 grupo modificado com otimizações no compilador para avaliar e comparar a redução do tempo de processamento sem o uso de otimizações.

Como na recarga são avaliados vários padrões de carregamento, também faz-se necessário calcular o tempo computacional para outras configurações desse benchmark, de modo a avaliar o comportamento da teoria de 1 grupo modificado para outras configurações do benchmark, considerando o tempo computacional médio para um dado número de configurações e a precisão da teoria de 1 grupo modificado aplicada aos cálculos neutrônicos de recarga. É importante ressaltar que objetivo aqui não é fazer um trabalho de otimização de recarga. Em vez disso, de redução do tempo computacional usando a teoria de 1 grupo modificado de energia aliadas à

otimizações no compilador do processador que, juntas, reduzem significativamente o tempo de processamento dos cálculos neutrônicos envolvido no processo de recarga. É necessário afirmar, no entanto, que os ganhos em termos de tempo computacional aqui demonstrados poderão ser usados em outras aplicações que versam sobre otimização de recarga.

No Capítulo 2, apresentam-se as duas teorias, de 2 grupos e 1 grupo modificado de energia.

No Capítulo 3, apresentam-se os conceitos de otimização do compilador, mostrando quais otimizações foram aplicadas no menu de propriedades do projeto e como cada uma delas influencia o tempo de processamento.

No Capítulo 4, apresentam-se os benchmarks que serão usados nas duas etapas deste trabalho e o gerador de núcleos, que será usado para gerar diferentes configurações embaralhando os ECs.

No Capítulo 5, apresentam-se os resultados correspondentes às etapas de validação, onde a viabilidade da teoria de 1 grupo modificado será demonstrada em termos dos desvios do k_{eff} , da densidade de potência e da redução do tempo computacional em relação à teoria de 2 grupos; e da análise de 1 ciclo completo, mostrando que os desvios e o tempo computacional estão coerentes com os resultados obtidos na validação, porém, considerando pesquisa de criticalidade, queima e depleção correspondente a 1 ciclo completo.

No Capítulo 6, apresentam-se as conclusões e discussões dos resultados obtidos.

Capítulo 2

Equação da Difusão de Nêutrons

2.1 Introdução

Este capítulo apresenta a equação da difusão de nêutrons em multigrupos de energia. Especificamente, a formulação com 1 grupo modificado a partir da teoria de 2 grupos de energia. À programação das equações de 1 grupo modificado deste capítulo serão aplicadas otimizações do compilador do programa Intel FORTRAN que servem para paralelização da CPU, melhorando o tempo de processamento computacional, como veremos no capítulo seguinte.

2.2 Considerações Iniciais

A obtenção do fluxo neutrônico e o fator de multiplicação efetivo de um reator nuclear consiste na solução de equação da difusão de nêutrons na formulação em multigrupos de energia. Esta equação é uma aproximação da equação de transporte de nêutrons (Aproximação P1) e, normalmente, é resolvida numericamente, fornecendo como resultado a distribuição do fluxo de nêutrons e o fator de multiplicação efetivo. Com a distribuição de fluxo é possível calcular a distribuição de potência no núcleo do reator para cada elemento combustível.

A solução numérica da equação da difusão de nêutrons, apesar de fornecer resultados com precisões muito boas, requer um alto custo de tempo computacional. As soluções numéricas para a equação da difusão de nêutrons na forma de multigrupos de energia são normalmente resolvidas para dois grupos de energia, térmico e rápido, resultando em duas equações. A equação da difusão de nêutrons com um grupo modificado reduz o problema para apenas uma única equação, que, no entanto, leva em conta os parâmetros nucleares dos dois grupos, permitindo obter uma redução considerável do tempo computacional envolvido e também a manutenção da precisão requerida para as soluções.

Na primeira seção deste capítulo, é apresentada a equação da difusão de nêutrons na forma de multigrupos e sua resolução com dois grupos de energia. Depois, será apresentada a resolução com um grupo modificado.

2.3 Equação da Difusão para 2 Grupos de Energia

A equação da difusão de nêutrons em sua formulação para dois grupos de energia, em duas dimensões, em geometria cartesiana, sem upscattering e independente do tempo, é escrita substituindo a Lei de Fick (Equação 2.2) na Equação da continuidade (Equação 2.1):

$$\sum_{u=x,y} \frac{\partial}{\partial u} J_{gu}(x, y) + \Sigma_{rg}(x, y)\phi_g(x, y) = \sum_{\substack{g'=1 \\ g' \neq g}}^2 \Sigma_{gg'}(x, y)\phi_{g'}(x, y) + \frac{\chi_g}{k_{eff}} \sum_{g'=1}^2 \nu \Sigma_{fg'}(x, y)\phi_{g'}(x, y); \quad g = 1, 2, \quad (2.1)$$

e a Lei de Fick é dada por:

$$J_{gu}(x, y) = -D_g(x, y) \frac{\partial}{\partial u} \phi_g(x, y); \quad u = x, y \quad (2.2)$$

onde $\phi_g(x, y)$ é o fluxo escalar de nêutrons do grupo g de energia, $J_{gu}(x, y)$ é a corrente líquida do grupo g de energia, $D_g(x, y)$ é o coeficiente de difusão do grupo g de energia, $\Sigma_{rg}(x, y)$ é a seção de choque macroscópica de remoção do grupo g de energia, $\nu \Sigma_{fg'}(x, y)$ é a seção de choque macroscópica de fissão multiplicada pelo número médio de nêutrons emitidos por fissão do grupo g' de energia, $\Sigma_{gg'}(x, y)$ é a seção de choque macroscópica de espalhamento do grupo g' para g , χ_g é o espectro de fissão do grupo g e k_{eff} é o fator de multiplicação efetivo.

Atribuindo 1 e 2 para o subscrito g , para os grupos de energia rápido e térmico, respectivamente, sendo os espectros de fissão do grupo rápido, $\chi_1 = 1$, e para o grupo térmico, $\chi_2 = 0$, quando substituirmos a Equação 2.2 na Equação 2.1 resultam duas equações: uma para o grupo rápido e outra para o grupo térmico.

$$-\frac{\partial}{\partial x} \left(D_1(x, y) \frac{\partial}{\partial x} \phi_1(x, y) \right) - \frac{\partial}{\partial y} \left(D_1(x, y) \frac{\partial}{\partial y} \phi_1(x, y) \right) + \Sigma_{r1}(x, y)\phi_1(x, y) = \Sigma_{12}(x, y)\phi_2(x, y) + \frac{1}{k_{eff}} \left[\nu \Sigma_{f1}(x, y)\phi_1(x, y) + \nu \Sigma_{f2}(x, y)\phi_2(x, y) \right], \quad (2.3)$$

$$-\frac{\partial}{\partial x} \left(D_2(x, y) \frac{\partial}{\partial x} \phi_2(x, y) \right) - \frac{\partial}{\partial y} \left(D_2(x, y) \frac{\partial}{\partial y} \phi_2(x, y) \right) + \Sigma_{r2}(x, y) \phi_2(x, y) = \Sigma_{21}(x, y) \phi_1(x, y). \quad (2.4)$$

Considerando-se que não há upscattering, isto é, $\Sigma_{12} = 0$. As seções de choque macroscópicas de remoção para os dois grupos ficam definidas como:

$$\Sigma_{r1}(x, y) = \Sigma_{a1}(x, y) + \Sigma_{21}(x, y) + D_1(x, y) B_z^2 \quad (2.5)$$

e

$$\Sigma_{r2}(x, y) = \Sigma_{a2}(x, y) + D_2(x, y) B_z^2, \quad (2.6)$$

onde $\Sigma_{a1}(x, y)$ e $\Sigma_{a2}(x, y)$ são as seções de choque macroscópicas de absorção dos grupos rápido e térmico, respectivamente. O termo B_z^2 representa o *Buckling axial*, que aparece devido à simplificação para o caso bidimensional.

2.4 Equação da Difusão para 1 Grupo Modificado

Para o caso de reatores de grande porte, assumimos que a fuga térmica é desprezível DUDERSTADT (1976), isto é, a formulação da equação da difusão para um grupo modificado é feita a partir das equações 2.3 e 2.4, então:

$$\frac{1}{\Sigma_{r2}(x, y) \phi_2(x, y)} \left[\sum_{u=x, y} \frac{\partial}{\partial u} J_{2u}(x, y) \right] \ll 1. \quad (2.7)$$

Logo, a equação (2.4) para o grupo térmico será reduzida para:

$$\phi_2(x, y) \cong \frac{\Sigma_{21}(x, y)}{\Sigma_{r2}(x, y)} \phi_1(x, y), \quad (2.8)$$

onde o fluxo do grupo térmico ($\phi_2(x, y)$) leva em conta os parâmetros nucleares dos dois grupos.

Substituindo a Equação (2.8) na Equação (2.3) obtemos a *Equação da Difusão para 1 Grupo Modificado*:

$$\sum_{u=x, y} \frac{\partial}{\partial u} J_{1u}(x, y) + \Sigma_{r1}(x, y) \phi_1(x, y) = \frac{1}{k_{eff}} \tilde{\Sigma}_f(x, y) \phi_1(x, y), \quad (2.9)$$

onde $\tilde{\Sigma}_f(x, y)$ é a *seção de choque de fissão efetiva*, definida como:

$$\tilde{\Sigma}_f(x, y) \equiv \nu\Sigma_{f1}(x, y) + \nu\Sigma_{f2}(x, y)\frac{\Sigma_{21}(x, y)}{\Sigma_{r2}(x, y)}. \quad (2.10)$$

Desprezando-se o termo de fuga do grupo térmico ocorre o desacoplamento das equações na formulação com dois grupos de energia, isto é, ao invés de duas equações diferenciais parciais de segunda ordem, uma única equação diferencial parcial de segunda ordem precisa ser resolvida, levando-se em conta os parâmetros nucleares dos dois grupos, e diminuindo o tempo computacional da solução numérica.

A solução numérica será baseada num artigo já publicado e que faz parte do desenvolvimento desta tese. Na publicação DA SILVA *et al.* (2023), discretizando a Equação (2.9) com esquema de integração centrado na malha e usando Gauss-Seidel como método de solução para o fluxo, foram comparados os resultados com as duas teorias, 2 grupos e 1 grupo modificado, para o k_{eff} e a distribuição de potência do reator Biblis, assim como os tempos de processamento para cada uma delas. O efeito das otimizações no compilador aplicadas à teoria de 1 grupo modificado também foi mostrado na redução do tempo de processamento em mais de 80%, sem alteração de nenhum dos resultados, afetando apenas o tempo de processamento. Nessa publicação também foram avaliadas 100 diferentes configurações do Biblis, para as quais foram calculados os valores de k_{eff} e seus respectivos desvios. A redução do tempo computacional em mais de 80%, proveniente das otimizações no compilador, também foi verificada para as 100 configurações do Biblis, mostrando que a metodologia de otimizações no compilador aplicadas à teoria de 1 grupo modificado foi bem-sucedida para redução dos tempos computacionais de cálculos de recarga.

O método de Gauss-Seidel permite resolver iterativamente o sistema linear pentadiagonal, atualizando os fluxos à medida que a solução avança nas regiões do núcleo do reator, reduzindo o tempo de processamento dos cálculos, bem como o número de iterações internas no fluxo médio de nêutrons, incluindo os tratamentos necessários para as regiões de contorno e de simetria, considerando um núcleo hipotético com $\frac{1}{4}$ de simetria. À programação da solução numérica da equação de difusão de um grupo modificada será submetida à otimizações no compilador Intel FORTRAN, que visam paralelizar a CPU, melhorando o tempo de processamento computacional, como veremos no capítulo seguinte.

Capítulo 3

Otimizações no Compilador

3.1 Introdução

Este capítulo é dedicado à introdução dos conceitos de otimização no compilador. As otimizações no compilador do sistema operacional são acionadas no menu de propriedades do projeto do Visual Studio, sem a necessidade de mudanças no código de programação (FORTRAN 90). As configurações do compilador Intel FORTRAN com sistema operacional Windows 10 podem ser selecionadas no menu de propriedades do projeto. Tratam-se basicamente de otimização da CPU, ou seja, da divisão de grandes tarefas computacionais, que são as simulações, para execução simultânea em vários processadores com o objetivo principal de reduzir o tempo computacional. São detalhadas todas as opções utilizadas nesta tese, permitindo ao leitor uma visão geral de como a paralelização, sem a necessidade de alterar qualquer linha do código original do cálculo neutrônico, pode afetar a performance das simulações computacionais.

3.2 Paralelização

A paralelização é uma das técnicas computacionais mais utilizadas para a redução do tempo computacional de problemas complexos. Ela consiste em dividir uma grande tarefa computacional em outras subtarefas menores que podem ser executadas simultaneamente em vários processadores ou núcleos, com o objetivo de melhorar o desempenho do código, a eficiência, ou até mesmo o tamanho do código, reduzindo o tempo computacional e aumentando a velocidade do processamento (BARNEY *et al.*, 2010) .

No caso dos problemas de física/engenharia nuclear onde simulações complexas exigem grande poder de processamento para serem concluídas em tempo hábil, a paralelização é o tipo de otimização mais utilizada para redução de tempo compu-

tacional. Atualmente, quase todo sistema de computação moderno utiliza alguma forma de paralelização. As próprias arquiteturas modernas já estão direcionadas a isso. No entanto, quando este tipo de arquitetura não é disponível ou se a performance computacional precisa ser melhorada ainda mais, a paralelização pode ser implementada explicitamente na programação, em regiões específicas do código, através de diretivas ao compilador, ou com instruções dadas a ele através das opções disponíveis de otimização dele próprio, sem que haja alteração nas linhas de código.

Este trabalho utilizou a paralelização sem alteração das linhas de código para cálculos neutrônicos. O objetivo foi melhorar o tempo computacional da teoria de 1 grupo modificado sem a necessidade de implementar programação explícita. Desta forma, garantindo a reprodutibilidade dos resultados, uma vez que não seria necessário para qualquer interessado ter acesso ao código, bastando apenas saber quais foram as opções de otimização utilizadas para paralelizar o código.

3.3 Otimizações no Compilador Intel

A função do compilador é executar uma série de etapas para converter o código-fonte de alto nível em código de máquina de baixo nível (PAHADE e DAWALE, 2019) , como mostra a Figura 3.3. Em relação às otimizações, estas abrangem várias técnicas e transformações que o compilador aplica ao código-fonte durante a compilação. Mas, por quê? Para melhorar o desempenho, a eficiência, e, em alguns casos, o tamanho do código de máquina resultante. Essas otimizações são fundamentais para influenciar vários aspectos da execução do código, incluindo velocidade, uso de memória e consumo de energia.

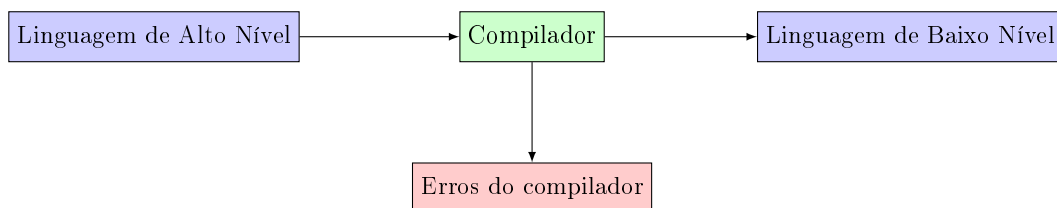


Figura 3.1: Esquema do Compilador.

Durante a compilação ocorrem etapas que envolvem: análise lexical, de sintaxe, semântica, geração de código intermediário, otimização e geração de código (PAHADE e DAWALE, 2019) , como mostra a Figura 3.2.

Durante a fase de otimização do código, o compilador busca meticulosamente maneiras de transformar um programa, visando uma saída semanticamente equivalente que utilize menos recursos ou execute mais rapidamente (PAHADE e DAWALE, 2019) . As opções de otimização empregadas neste processo abrangem, mas não se limitam em: simplificação de estruturas, otimização de loops, inlining de função,

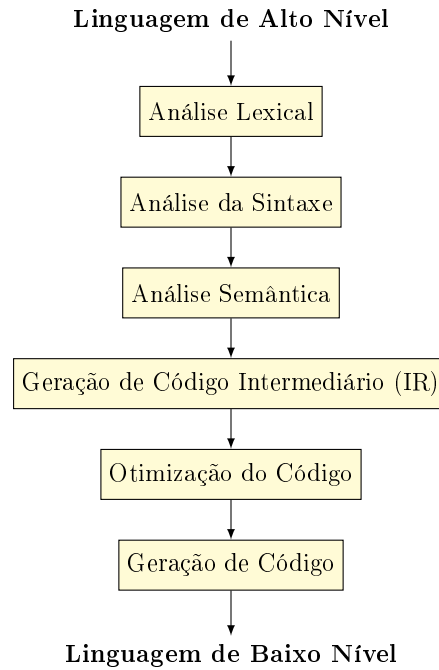


Figura 3.2: Estágios da Compilação.

eliminação de código redundante (ou morto), otimização do uso da memória (neste caso, da memória cache) e outras específicas da máquina (MOGENSEN, 2009). Esta última leva em consideração os recursos específicos do processador ou arquitetura da máquina alvo (hardware) para aprimorar ainda mais o desempenho. As otimizações suportadas pelo hardware, como veremos adiante, são cruciais para compreender o conceito de *Multithreading*.

Neste capítulo, serão discutidas as opções de otimização que melhoram o desempenho do código quando acionadas pelos *check boxes* do menu de propriedades do projeto *Fortran Parallel Studio*. As opções de otimização podem ser encontradas em INTEL (2018). Neste trabalho não foi utilizada programação paralela, de modo que o próprio compilador aplica as otimizações ao código, isto é, não houve qualquer tipo de controle direto através de cláusulas de diretivas de programação paralela. As subseções seguintes discorrem sobre cada uma das opções de otimização utilizadas neste trabalho e como cada uma delas podem ser acionadas. Uma subseção separada discorre sobre OpenMP. E no Apêndice C, são mostrados prints de todas as opções de otimização utilizadas neste trabalho.

3.3.1 Opções de Otimização

A) Nível de Otimização

Os níveis de otimização disponíveis são:

- /Od (Disable): Desativa otimizações no código por completo e melhora in-

formações de depuração. No entanto, algumas aplicações não funcionarão corretamente sem que opções de otimização mais avançadas sejam ativadas. O processo de depuração, *debugging* ou *debug*, é aquele que serve para encontrar e reduzir defeitos num aplicativo de software ou mesmo em hardware. Erros de software incluem aqueles que previnem o programa de ser executado e aqueles que produzem um resultado inesperado. Esta opção não é recomendada senão para propósitos de depuração;

- /O1 (Minimize Size): Otimiza a velocidade evitando o aumento do tamanho do código. Adequado para aplicativos com tamanhos de código grandes, muitas ramificações e onde o tempo de execução não é dominado por código dentro de loops. É o nível de otimização mais baixo;
- /O2 (Maximize Speed): Define as otimizações que geram o código mais rápido na maioria dos casos. É o nível mais recomendado de otimizações (default), a não ser que o problema exija otimizações mais avançadas. Melhorias em relação a /O1: ativa a vetorização;
- /O3 (Maximize Speed plus Higher Level Optimization): O maior nível de otimização possível. Recomendado para aplicativos com cálculos de ponto flutuante com muitos loops e grandes conjuntos de dados. Ele ativa otimizações que são custosas em termos de tempo de compilação e uso de memória. Este nível de otimização pode resultar em alterações funcionais no programa se ele for sensível a exceções de ponto flutuante, ao sinal de zero ou aos efeitos de precisão da reordenação de cálculos. Apesar de muito utilizada, otimizações do tipo -O3 não garantem a melhoria da performance, e, na verdade, em muitos casos podem até diminuir a performance do sistema por causa de binários maiores e maior uso de memória;
- Custom (Customizada): Nesta opção o usuário não especifica diretamente quais as otimizações devem ser aplicadas no compilador, deixando a critério do próprio “decidir” qual a melhor opção.

B) Vetorização

Esta otimização é acionada automaticamente quando o nível de otimização definido for /O2 ou /O3. A vetorização é um caso especial de paralelização automática, onde um programa de computador é convertido de uma implementação escalar, que processa um único par de operandos por vez, para uma implementação vetorial, que processa uma operação em vários pares de operandos de uma só vez, como mostra a Figura 3.3 (EIGENMANN e HOEFLINGER, 2000) . Isto é possível através da

instrução **SIMD**, sigla para *Single Instruction Multiple Data*, para arquiteturas de computadores que permitem paralelização, onde múltiplos cálculos são processados com apenas uma única instrução. A instrução SIMD é implementada através das extensões **AVX** e **SSE**, siglas para *Advanced Vector Extension* e *Streaming SIMD Extension*, respectivamente (BIK *et al.*, 2008). As extensões SSE (nas versões SSE4.1 e SSE4.2, opções que serão importantes na geração de código) e AVX, consistem em dois conjuntos de instruções com registradores de 128 e 256 bits, respectivamente.

Durante a compilação, a vetorização transforma esses loops procedurais em sequências de operações vetoriais. Isto é, transforma esses loops atribuindo uma unidade de processamento a cada par de operandos. Os programas passam a maior parte do tempo dentro desses ciclos. Portanto, a vetorização pode acelerá-los significativamente, especialmente em grandes conjuntos de dados. Uma vez ativada, a vetorização exigirá que outras opções sejam acionadas.

Vetorização Automática - SIMD

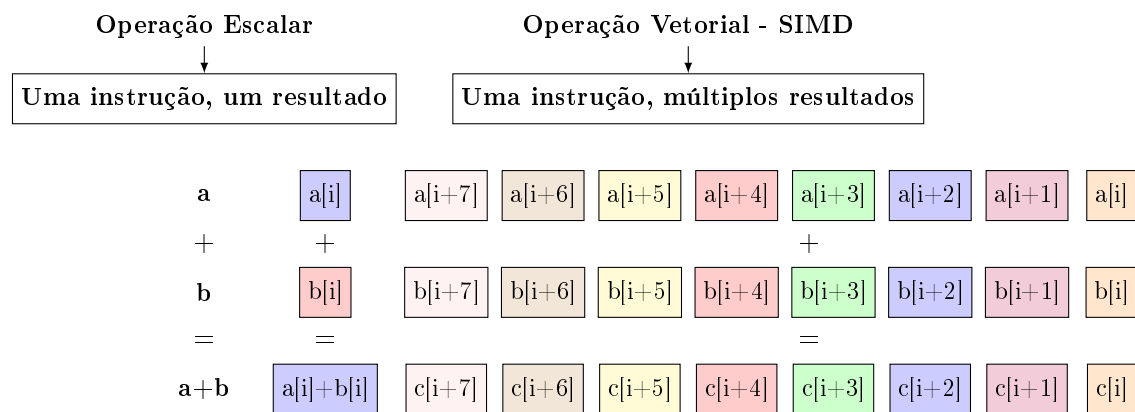


Figura 3.3: Esquema da vetorização automática.

C) Multiprocessadores

A opção de otimização para compilação em multiprocessadores está relacionada à arquitetura da máquina. Ao acionar a opção `/MP`, reduz-se o tempo total de compilação dos arquivos de origem na linha de comando, fazendo com que o compilador crie uma ou mais cópias de si mesmo, cada uma em um processador separado. Em seguida, essas instâncias compilam simultaneamente os arquivos de origem. Em alguns casos, o tempo total para construir os arquivos de origem pode ser reduzido significativamente.

D) Função *Inlining*

Funções *Inline* são diretivas do compilador para a substituição do corpo da função

na instância de cada chamada dessa função, ou seja, economizando assim a sobrecarga da função (**call**), tornando o programa mais rápido (BERGMANN, 2017) . As opções de otimização dessas funções são do tipo `"/Ob"`, e podem ser:

- `/Obd` ou `/Ob0` (Disable): Desabilita este tipo de otimização;
- `/Ob1` (Only Inline Directive): Permite apenas o *Inline* de funções marcadas explicitamente como **inline**, `__inline` ou `__forceinline`;
- `/Ob2` (Any Suitable): Permite que o próprio compilador decida se é benéfico ou não substituir uma função não marcada explicitamente pelo seu corpo no lugar de sua chamada.

O compilador trata este tipo de otimização como sugestão. Não há garantia que sejam atendidas. É possível desabilitar este tipo de atualização, mas não é possível forçar o compilador para otimizar funções, mesmo que marcadas com a palavra-chave `__forceinline`.

E) Tamanho ou Velocidade do Código

As opções de otimização para o compilador em relação ao tamanho ou velocidade do código são acionadas por `/Os` e `/Ot`, respectivamente. As duas opções são definidas como:

- `/Os` (Favor Small Code): Minimiza o tamanho dos executáveis (exe.) e biblioteca de dados (dll - *Dynamic-link library*) instruindo o compilador a favorecer a redução do tamanho em detrimento da velocidade (INTEL, 2024) . As opções `/Os` e `/Ot` permitem especificar a preferência por uma em detrimento à outra;
- `/Ot` (Favor Fast Code): Maximiza a velocidade dos executáveis e biblioteca de dados instruindo o compilador a favorecer o aumento da velocidade em detrimento da tamanho. Esta é a opção *default* quando as opções de otimização estão desabilitadas. Esta opção é implícita quando a opção `/O2` é habilitada; a otimização (`/O2` - Maximize Speed) combina diversos tipos de opções para acelerar o código.

F) Paralelização Automática

A Paralelização Automática (Parallelization) é acionada com `/Qparallel`. Paralelismo é uma forma de computação em que vários cálculos são realizados ao mesmo

tempo, sob o princípio de que grandes problemas podem ser divididos em pequenas partes, que são resolvidos concorrentemente em paralelo (MELHEM, 1992) .

Os laços de iteração (comandos **for** ou **while**, por exemplo) são a principal fonte de paralelismo, principalmente se as iterações do laço são independentes, de tal forma que podemos dividir as iterações entre threads. A paralelização automática gera threads em loops executados em paralelo. O conceito de thread será analisado na próxima seção.

Normalmente, o paralelismo geralmente é feito de forma explícita na programação do código, embora já existam computadores paralelos cujo hardware suporta paralelismo, isto é, não necessitam de programação paralela, ficando implícita quando comandos que são fontes de paralelização são utilizados. As linguagens de programação paralelas de amplo uso continuam explícitas, isto é, requerendo a programação paralela propriamente dita; ou no máximo, parcialmente implícitas, quando o programador fornece ao compilador diretivas para o paralelismo, que é o caso da otimização.

G) Pré-carregamento de Cache

Esta opção diz respeito à otimização da memória cache. O pré-carregamento (ou pré-busca) de cache é uma técnica usada pelo processador do computador para aumentar o desempenho de execução, buscando instruções ou dados no seu armazenamento original (endereço de memória cache) antes do seu efetivo uso (daí o termo ‘pré-busca’). A maioria dos processadores de computador modernos possui memória cache local e rápida, na qual os dados pré-buscados são mantidos até serem necessários. A fonte para operações de pré-busca geralmente é a memória principal (RAM). Devido ao seu design, o acesso às memórias cache é normalmente muito mais rápido do que o acesso à memória principal, portanto, pré-buscar os dados e acessá-los dos caches é geralmente muito mais rápido do que diretamente da memória principal. A pré-busca pode ser feita com instruções de controle de cache sem bloqueio, isto é, gerencia a memória cache sem interromper a execução do programa

A otimização para o pré-carregamento (*Prefetch Insertion*) é acionada através de */Qopt-prefetch : n*, onde *n* pode assumir valores de 0 a 3. Os níveis mais baixos de *n* reduzem a quantidade de pré-busca, sendo *n = 0* para desabilitar otimização (**Disable**); *n = 2* (**Medium**) como *default*; e *n = 3* (**Agressive**) a opção que maximiza a otimização. Para *n = 3*, a opção requer nível de otimização */O3* no compilador, para compiladores com configuração */QxSSS3* ou superior, que é o caso dos processadores Intel.

H) Otimização Interprocedural

qualquer estrutura de dados que representam o programa sem perda de informação, de tal forma que a execução seja precisa. A IR serve como uma interface entre o compilador e seus componentes. Pelo fato de ser interna ao compilador, cada um é livre para definir a forma e os detalhes de suas IRs. A IR existe apenas durante o processo de compilação, ou podem ser arquivos de saída do tipo texto ou binário. A importância da IR é que ela deve ser o mais geral possível para ter capacidade de representar programas em múltiplas linguagens, isto é, independente da máquina (BERGMANN, 2017) .

Para melhor compreensão do processo de compilação **LTO**, o esquema da Figura 3.5 mostra a divisão desse processo com o que se conhece como **middle end**, destacado na cor cinza. Nesta etapa, o compilador aplica diferentes transformações na IR, de tal maneira que o programa possa ser mais eficiente. Na Figura 3.5, a Tabela de Símbolos é destacada devido à sua utilização em todas as fases da compilação para armazenar e gerenciar informações sobre os identificadores (símbolos) presentes no código-fonte.

A compilação envolve uma gradual redução da representação do nível mais alto de construtores de programação (**front ends**, relacionados aos aspectos da programação) para o mais baixo (**back ends**, relacionados aos aspectos do hardware da máquina). O linguagem de baixo nível, ou o código de máquina, é geralmente mais longo porque reflete os detalhes da máquina onde o programa é executado. O que se denomina linguagem de baixo nível é aquela que inclui informações suficientes para permitir a execução correta do programa. Por isso, para que a IR seja capaz de representar múltiplas linguagens, a IR precisa de ser mais próxima da linguagem de máquina para executar todas as plataformas. O compilador deve suportar múltiplas linguagens, traduzindo-as para múltiplas plataformas (na verdade, arquiteturas diferentes).

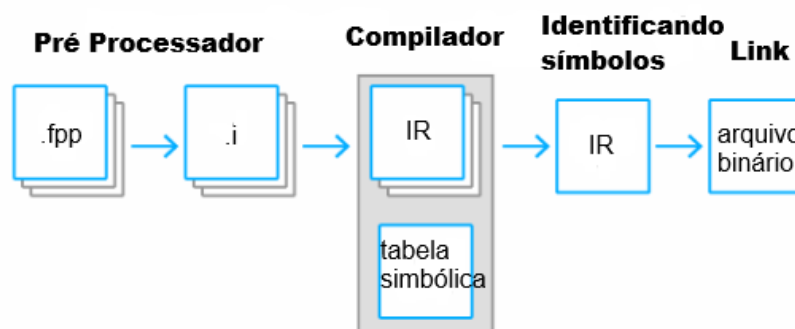


Figura 3.5: Representação Intermediária - IR.

Em muitos casos, as otimizações que podem ser aplicadas estão relacionadas às arquiteturas específicas. Neste caso, a otimização IPO oferece dois modelos de

compilação: em arquivo único, pela opção **Single-file:/Qip**, ou em vários arquivos, pela opção **Multi-file:/Qipo**. A compilação em arquivo único resulta em um único arquivo objeto formado a partir do arquivo normal. A compilação em vários arquivos resulta em um ou mais arquivos objetos simulados. Os arquivos de objetos simulados contêm o IR, em vez do código de objeto normal. Os arquivos de objetos simulados podem ser dez vezes maiores e, em alguns casos, maiores que o tamanho dos arquivos objetos normais. Além disso, o compilador coleta informações dos arquivos de origem individuais que compõem o programa. Usando essas informações, o compilador realiza otimizações em funções e procedimentos em diferentes arquivos de origem. Inlining é a otimização mais poderosa suportada pelo IPO.

I) Pré-Processamento

O papel de pré-processador, como já mencionado na LTO, é interno ao sistema e anterior à compilação, como na Figura 3.4. Muitos compiladores FORTRAN, incluindo GNU FORTRAN, permitem passar o código-fonte através de um pré-processador C (CPP; às vezes também chamado de pré-processador FORTRAN, FPP - neste caso, /fpp) para permitir a compilação condicional. No caso do GNU Fortran, este é o pré-processador GNU C no modo tradicional. Em sistemas com nomes de arquivos com preservação de maiúsculas e minúsculas, o pré-processador é invocado automaticamente se a extensão do nome do arquivo for: **.F**, **.FOR**, **.FTN**, **.fpp**, **.FPP**, **.F90**, **.F95**, **.F03** ou **.F08**.

J) Geração do Código

A geração do código (**Code Generation**) diz respeito à fase final do processo de compilação, onde o compilador converte a representação intermediária (IR) do código fonte em linguagem de máquina (MOGENSEN, 2009) . A opção /Qax é acionado para otimizar o código intermediário com base em extensões SIMD específicas e criar o executável. As instruções SIMD (como vimos na Vetorização) estão relacionadas à arquitetura da máquina na qual o código será compilado. Ao otimizar o IR, a opção /Qax permite que um único binário tenha desempenho máximo em uma variedade de processadores diferente. Um argumento junto à opção /Qax indica a extensão das instruções SIMD da Intel® (Intel® SSE), SSSE e AVX, como vimos também na Vetorização. Neste trabalho /QaxSSE4.2 Otimiza para instruções SSE4.2.

O caminho do código gerado é determinado pela arquitetura especificada acionando as opções /Qx (sistema operacional Windows). Ao especificar as opções /Qax e /Qx, o código gerado só será executado em processadores Intel® compatíveis com a configuração especificada para /Qx, forçando a otimização apenas para a arquitetura

tura mais alta especificada e desconsiderando a compatibilidade com processadores mais antigos.

L) Formato do Arquivo

O formato do arquivo fonte (**Source File Format**, o arquivo normal) a ser compilado também deve ser considerado antes da compilação. A opção **/fixed**, para Formato Fixo (**Fixed Format**), é acionada para versões anteriores ao Fortran 90; a opção **/free**, utiliza o Formato Livre (**Free Format**) para versões 90, 95 e 2003 do Fortran. A opção *default*, File Extension, escolhe a extensão do arquivo fonte que é reconhecida pelo compilador.

M) Diretivas para OpenMP

A opção **/Qopenmp** aciona as diretivas OpenMP, responsáveis pela implementação do modelo de paralelização *multithreading* que será discutido com mais detalhe na seção a seguir.

Por fim, depois de acionadas todas as opções e seus respectivos switches, uma única linha de comando é gerada, mostrando todas as otimizações que serão aplicadas ao código.

3.4 OpenMP

3.4.1 Introdução

OpenMP (ou Open Multi-Processing) é uma interface de programação de aplicativo (*application programming interface* - API) para a programação paralela multiprocesso de memória compartilhada em múltiplas plataformas como Linux, macOS e Windows. Consiste num conjunto de diretivas de compilador, rotinas de biblioteca, e variáveis de ambiente que influenciam o comportamento do tempo de execução (CHANDRA, 2001). Diretivas são linhas especiais de código fonte com significado especial apenas para os compiladores. Podem ser feitas diretamente na programação explícita, ou através do menu de propriedades do programa, onde as opções de otimização são selecionadas sem a necessidade de alteração no código.

Acompanhando o desenvolvimento tecnológico dos computadores, inicialmente, as diretivas de programação paralela para memória compartilhada não eram padronizadas. Cada fabricante tinha seu próprio conjunto de diretivas. Então, à medida que o desenvolvimento dos computadores foi avançando, o fórum OpenMP foi iniciado pela Digital, IBM, Intel, KAI e SGI, a fim de reunir os principais fabricantes de hardware e software para padronizar um conjunto de diretivas de programação

paralela para memória compartilhada. Atualmente, OpenMP é administrado pelo *OpenMP Architecture Review Board* (ou *OpenMPARB*). Em Outubro de 1997, o padrão OpenMP para Fortran foi liberado. A versão 2.0 para Fortran foi liberada em Novembro de 2000. O padrão OpenMP para C/C++ foi liberado em Outubro de 1998. E a versão 2.0, liberada em Março de 2002.

Para melhor compreensão do funcionamento do OpenMP é necessário também relacioná-lo à arquitetura do computador a à implementação do Modelo de Multithreading, que nada mais é que a distribuição de tarefas entre os processadores por meio das threads.

3.4.2 Compartilhamento de Memória

Os computadores podem ser classificados quanto à arquitetura como: sequenciais, de memória distribuída e centralizada. Esta última está relacionada diretamente ao funcionamento do OpenMP, pois a maneira que os processadores acessam a memória influi no desempenho computacional. Um melhor entendimento dos tipos de arquitetura pode ser encontrado no artigo BARNEY *et al.* (2010) .

- **Memória Compartilhada:** Neste tipo de arquitetura, vários processadores acessam uma memória compartilhada, isto é, compartilham um único espaço de memória, no qual cada processador pode ler e escrever em todas as posições de memória. Este esquema é conhecido como **Multiprocessador**. A Figura 3.6 mostra vários processadores, representados pela letra **P**, interconectados com a memória por um barramento único. Este esquema também é conhecido por **Memória Compartilhada Centralizada (UMA - Uniform Memory Access)**, também conhecida pela sigla SMP, de *Symmetric Multiprocessing*, que consiste em dois ou mais processadores idênticos conectados a uma única memória global que encontra-se localizada a mesma distância de todos os processadores. Este tipo possui melhor comunicação entre as unidades, pois permite que qualquer processador trabalhe em qualquer tarefa, não importando onde ele esteja localizado, otimizando a carga de trabalho; tempo de acesso à memória igual para todos os processadores; e nós de processamento e memória interligados através de barramento único.
- **Memória Distribuída:** Neste tipo de arquitetura, cada processador “enxerga” sua própria memória. Não há mais um único espaço de endereçamento, há espaços de endereçamento privados, cada um referente a cada memória ligada a seu respectivo processador. A Figura 3.7 mostra este esquema deste tipo de compartilhamento de memória, também conhecido como **Multicomputador**. Neste caso, através de uma rede de alta velocidade interconectada,

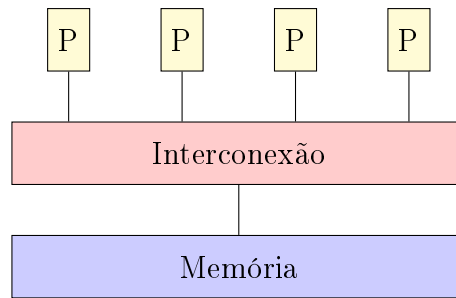


Figura 3.6: Esquema de Memória Compartilhada Centralizada.

um sistema de troca de mensagens e dados é utilizado quando um processador necessita de dados de outro processo que ele próprio não esteja executando. Isto é possível através de operações do tipo **send** e **receive**. Este tipo possui melhor flexibilidade em relação ao número de unidades de processamento, permitindo dimensionar o sistema de acordo com o problema a ser tratado, além do fato de que uma possível falha em um nó de processamento não afetaria a indisponibilidade de todo o sistema. Este tipo de arquitetura não suporta OpenMP. No entanto, há outras arquiteturas que possuem características dos dois tipos, compartilhada e distribuída, como veremos a seguir:

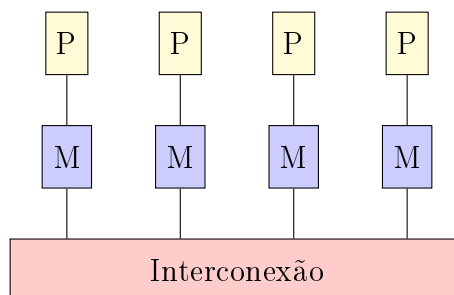


Figura 3.7: Esquema de Memória Distribuída.

- **NUMA (*Non-Uniform Memory Access*)**: neste tipo, a memória é dividida em tantos blocos quanto forem os processadores, e cada bloco de memória é conectado a um processador com memória local, como mostra a Figura 3.8. Neste tipo de arquitetura, o acesso aos dados que estão na memória local (em cada bloco) é mais rápido que o acesso aos dados em outros blocos, isto é, existem distâncias diferentes da memória. Isso significa, que o tempo de acesso à memórias locais é menor que à memórias remotas. Apesar de, aparentemente, a memória estar distribuída, com cada bloco associado a um processador, o espaço de endereçamento da memória ainda é único;

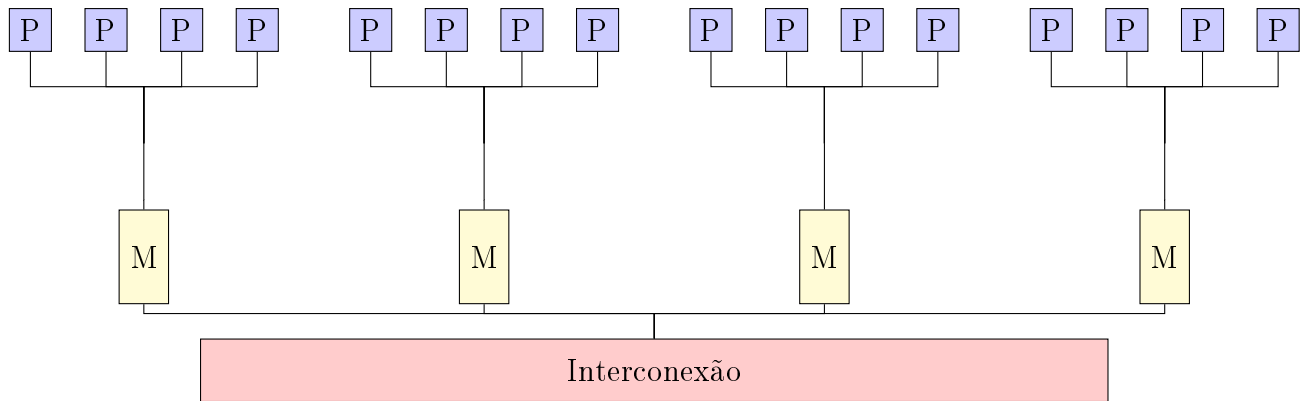


Figura 3.8: Esquema do NUMA.

- **COMA (*Cache-Only Memory Architecture*)**: também chamadas de **COMA Caches**, assemelha-se à arquitetura NUMA, com a diferença que há uma memória cache para cada processador. A Figura 3.9 mostra cada processador (letra **P**) ligado a um bloco de memória cache (letra **C**), isto é, todas as memórias locais de cada bloco estão estruturadas como memórias cache. O cache é um bloco de memória para o armazenamento temporário de dados que possuem uma grande probabilidade de serem utilizados novamente. A principal vantagem na utilização do cache consiste em evitar o acesso ao dispositivo de armazenamento (RAM) - que pode ser demorado - armazenando os dados em meios de acesso mais rápidos. O OpenMP, por ser uma plataforma de programação paralela de memória compartilhada, necessita de grandes blocos de cache, uma vez que no sistema de memória compartilhada vários processadores necessitam acesso rápido à memória e por isso utilizam sistemas de cache. Caso contrário, o tempo de acesso à memória RAM seria muito grande e não haveria ganho de tempo substancial. O cache também tem papel fundamental para o funcionamento das threads, como veremos na seção seguinte.

3.4.3 Threads

As threads são uma consequência da programação paralela. No caso do OpenMP, as threads são geradas quando a otimização Diretivas para OpenMP (item M da seção anterior) é acionada pela opção `/Qopenmp`. No entanto, a otimização Paralelização Automática (item F da seção anterior), acionada pela opção `/Qparallel`, que também serve para a paralelização automática de blocos de repetição (loops) que podem ser executados em paralelo por meio de threads. É por esta razão que

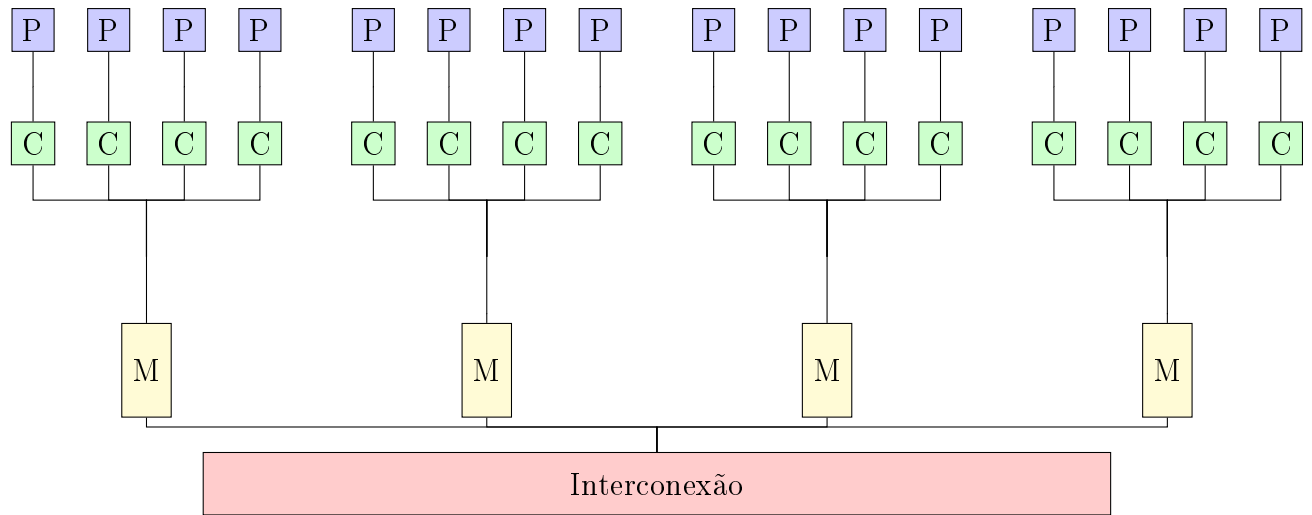


Figura 3.9: Esquema do COMA.

OpenMP é também conhecido por **Implementação de Multithreading**, que é a execução em massa de várias threads em um mesmo processo. A Figura 3.10 mostra como o OpenMP implementa as threads. Basicamente, consistem em pequenos programas que trabalham como um subsistema cuja função é dividir processos em duas ou mais tarefas, originando múltiplos caminhos de execução concorrentes, onde cada uma das threads executa uma tarefa diferente (CHANDRA, 2001) .

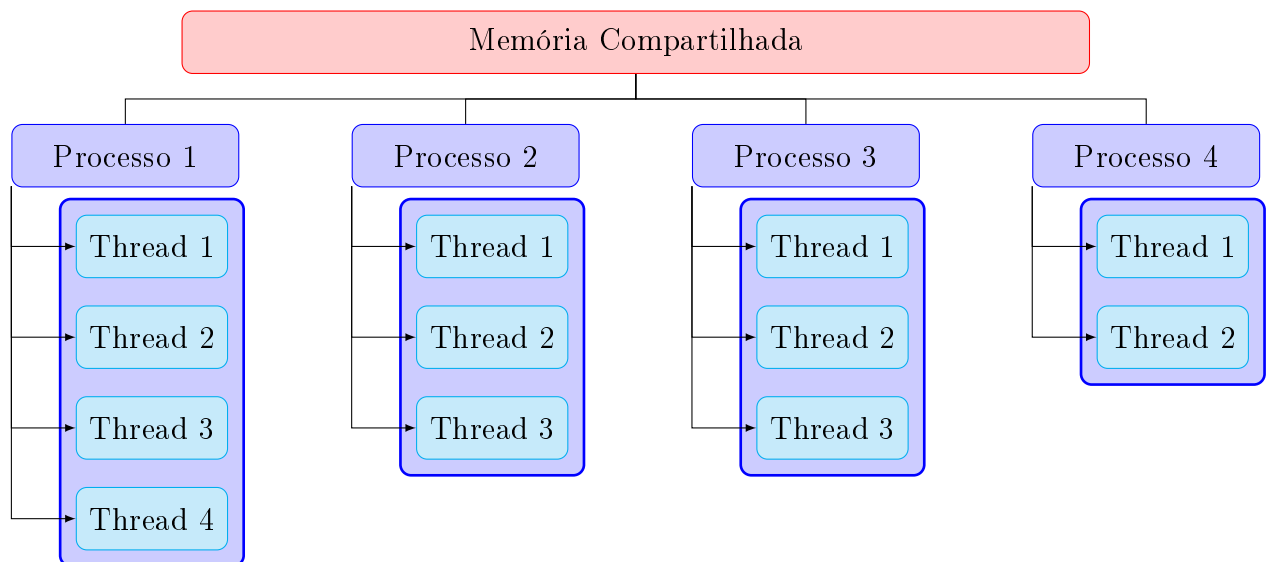


Figura 3.10: Implementação de Multithreading.

Um processo consiste em uma aplicação sendo feita. Ele usa uma parte da memória do sistema e não pode ser compartilhado com outro processo. As threads podem compartilhar dados com outras threads, mas cada uma delas tem dados

privados. A comunicação se dá através de uma área de memória compartilhada, onde as threads podem ler ou escrever nas variáveis compartilhadas. Pode-se considerar uma thread também como um “processo leve”, e por consumir menos recurso, quando há multithreading, é possível afirmar que o sistema possui ainda mais performance, sempre possibilitando que as threads sejam executadas de maneira independente e compartilhando recursos simultaneamente.

O conceito de multithreading pode ser melhor compreendido quando entendemos como as threads trocam informações entre si dentro de uma região paralela (CHANDRA, 2001) . Primeiro, é preciso definir o que é uma região paralela, e depois, como as threads trocam informações entre si.

I - Regiões Paralelas/Modelo de Execução Fork-Join

A região paralela é a estrutura básica de paralelismo no OpenMP. Ela define uma seção do programa. Em relação às threads, pode-se dizer que é uma seção ou bloco do programa executado por todas as threads simultaneamente. Dentro de uma região paralela, um código é executado por todas as threads, e durante a execução do programa, uma única thread é criada e executada serialmente, denominada **master thread**. No momento que ela encontra uma região paralela, com uma instrução OpenMP, “cria” uma equipe de threads que dividem uma tarefa na região. A Figura 3.11 mostra três regiões paralelas, com diferentes números de threads. Este modelo de execução de tarefas é chamado de **Modelo de Execução Fork-Join**. Ao encontrar uma região paralela (ponto vermelho), a master thread bifurca (do inglês, *forks*) em várias threads que executam em paralelo tarefas distribuídas uniformemente pela master thread. Ao final de cada região paralela, encontra-se uma espécie de barreira implícita, que serve como ponto de referência onde cada thread esperará ao término de sua tarefa. Quando todas as threads terminarem suas tarefas elas saem juntas pela barreira (do inglês, *join*). Neste momento só resta a master thread, e ela executa o código serialmente depois de passar pela região paralela.

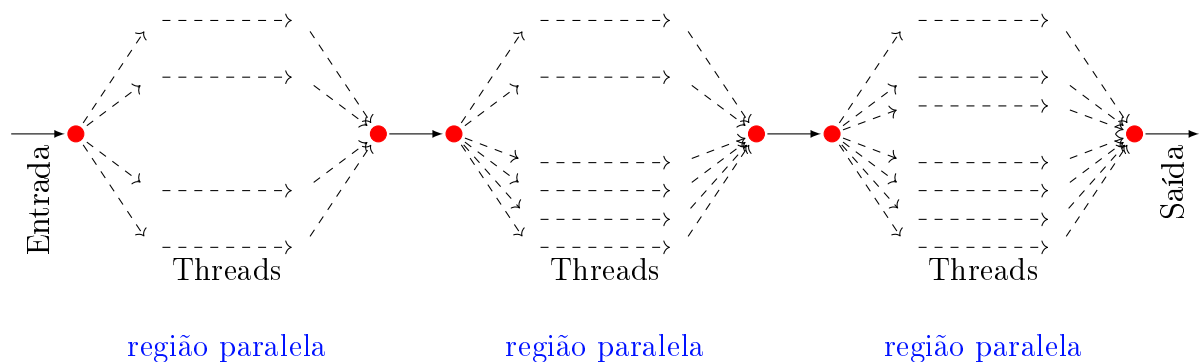


Figura 3.11: Modelo de Execução Fork-Join.

Para entender como as threads trocam informações entre si precisamos explorar os tipos de dados trocados entre elas (as variáveis privadas e compartilhadas entre elas) e como ocorre a sincronização, ou seja, como as threads escrevem e leem as variáveis compartilhadas.

II - Dados Privados e Compartilhados

Dentro da região paralela as variáveis podem ser privadas ou compartilhadas. Todas as threads veem a mesma cópia das variáveis compartilhadas, permitindo ler ou escrever nas variáveis compartilhadas. No entanto, cada thread tem suas variáveis privadas, que são invisíveis para as outras threads. A variável privada pode ser lida ou escrita apenas por sua própria thread, como índices de laços de iteração, por exemplo. A classificação do tipo de variável, se privada ou compartilhada, é feita através de cláusulas (**clauses**). As cláusulas são informações adicionais nas diretivas em regiões paralelas e se distinguem pela existência de uma sentinela no começo da linha ¹. Por exemplo, as sentinelas do OpenMP são:

- **FORTRAN**: `!$OMP` (ou `C$OMP` ou `*$OMP`);
- **C/C++**: `#pragma omp`.

E as cláusulas **SHARED**, para variável compartilhada, e **PRIVATE**, para variável privada, são assim declaradas:

- **FORTRAN**: `!$OMP parallel [clause];`
- **C/C++**: `#pragma omp parallel [clause];`

e, no lugar de “[clause]”, escreve-se: `shared` ou `private`.

III - Sincronização

Uma vez que as threads podem ler e escrever em variáveis compartilhadas é necessário sincronizar estas ações, de forma que também seja necessário a ordenação correta de leitura e escritas nas variáveis compartilhadas. A atualização dessas variáveis devem ser protegidas de tal modo que uma thread só pode escrever ou ler uma variável se ela não estiver sendo escrita ou lida por outra variável. A sincronização em si, funciona através de mecanismos que controlam o acesso a recursos compartilhados, como variáveis compartilhadas e memória, garantindo que apenas uma thread por vez possa acessar uma região paralela do código. Desta

¹A aplicação deste conceito é melhor compreendida quando se discute programação paralela, onde o próprio programador define as regiões a serem paralelizadas e indica as cláusulas das diretivas.

forma, evita-se condições de corrida e corrupção de dados, utilizando técnicas como bloqueios (**lock** - trancar), métodos de espera/notificação (**wait()**/**notify()**) e semáforos (**sinalizadores**). Quando uma thread acessa um recurso sincronizado, ela “bloqueia” o acesso para outras, liberando-o apenas ao terminar, enquanto outras threads podem “esperar” ou serem “notificadas” para continuar.

Capítulo 4

Apresentação dos Benchmarks e do Gerador de Núcleos

Este capítulo é dedicado à apresentação dos benchmarks que foram utilizados neste trabalho para mostrar a viabilidade da teoria de 1 grupo modificado nos cálculos de otimização de núcleo. Três benchmarks foram usados: um de reator de grande porte, **Biblis**, um de pequeno porte, **Koebke**, e outro que foi usado na análise de 1 ciclo completo, **Benchmark 19 (Two-dimensional PWR-Burnup Problem - with Fuel Management)**. O Biblis foi usado apenas na etapa de validação, onde a distribuição de potência e o fator de multiplicação efetivo são calculados usando as duas teorias, e depois, com de 1 grupo modificado otimizado, avaliando o tempo computacional para cada uma. O Koebke foi usado para avaliar se a teoria de 1 grupo modificado, mesmo sendo restrita a reatores de grande porte, produziria resultados dentro da mesma margem de erro quando aplicada a reatores de grande porte. O terceiro benchmark foi usado na análise de 1 ciclo completo, fazendo pesquisa de criticalidade, queima e depleção usando as duas teorias, e depois, com 1 grupo modificado otimizado. Os resultados obtidos mostrando as discrepâncias da teoria de 1 grupo modificado em relação à de 2 grupos estão no Capítulo 5.

A primeira seção deste capítulo descreve o benchmark Biblis destacando suas dimensões principais e tipos. A seção seguinte descreve a configuração e as dimensões principais do benchmark Koebke. A terceira seção descreve o benchmark que será usado na análise de um ciclo completo: pesquisa de criticalidade, queima e depleção. A última seção descreve o gerador de núcleos utilizado para gerar diferentes configurações que serão simuladas tanto na etapa de validação, para os benchmarks Biblis e Koebke, quanto na etapa de análise de um ciclo completo, para o benchmark de pesquisa de criticalidade, queima e depleção. Esse gerador de núcleos é baseado no Modelo de Chaves Aleatórias e será utilizado para realizar os cálculos para 100 configurações, embaralhando os elementos combustíveis sem restrição quanto à posição,

isto é, sem levar em consideração as posições do ECs na simetria do núcleo.

4.1 Benchmark Biblis

A Figura 4.1 mostra o núcleo do benchmark Biblis com simetria de 1/4, com as dimensões principais dos elementos combustíveis destacadas e os oito tipos que compõem o núcleo. Os parâmetros nucleares são apresentados na Tabela 4.1. O Biblis é um reator de potência, tipo PWR, onde o “baffle” e o refletor são homogeneizados em uma única região (tipo 8) (MÜLLER e WEISS, 1991) . Foi utilizada a condição tipo vácuo no contorno.

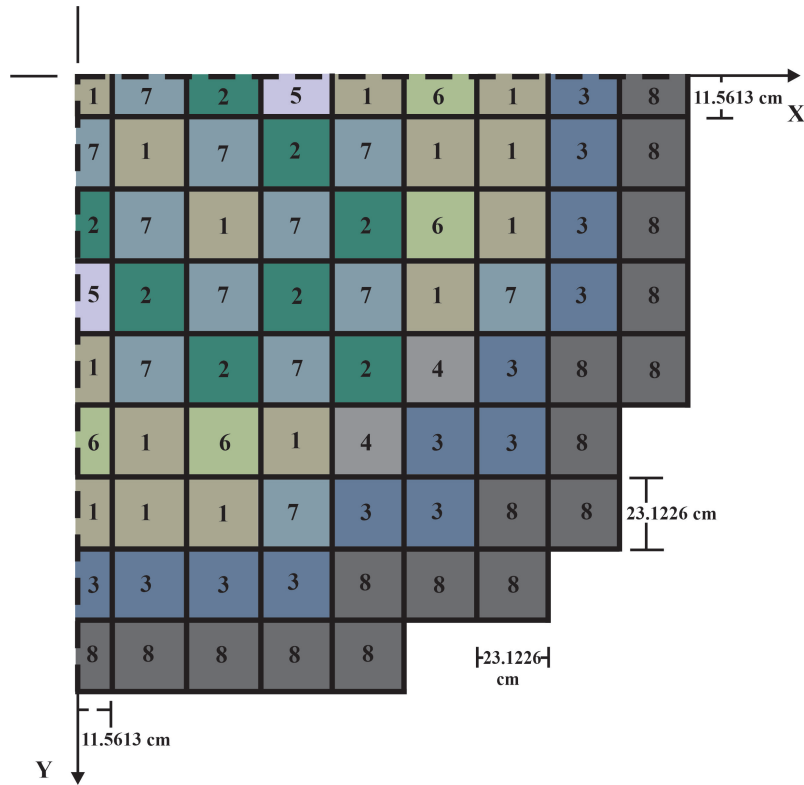


Figura 4.1: Núcleo do benchmark Biblis com 1/4 de simetria.

Tabela 4.1: Parâmetros nucleares - Biblis.

Tipo	g	$\Sigma_{ag}(cm^{-1})$	$\nu\Sigma_{fg}(cm^{-1})$	$D_g(cm)$	$\Sigma_{gg'}(cm^{-1})$
1	1	0,0095042	0,0058708	1,4360	0,000000
	2	0,0750580	0,0960670	0,3635	0,017754
2	1	0,0096785	0,0061908	1,4366	0,000000
	2	0,0784360	0,1035800	0,3636	0,017621
3	1	0,0103630	0,0074527	1,4389	0,000000
	2	0,0914080	0,1323600	0,3638	0,017101
4	1	0,0100030	0,0061908	1,4381	0,000000
	2	0,0848280	0,1035800	0,3665	0,017290
5	1	0,0101320	0,0064285	1,4385	0,000000
	2	0,0873140	0,1091100	0,3665	0,017192
6	1	0,0101650	0,0061908	1,4389	0,000000
	2	0,0880240	0,1035800	0,3679	0,017125
7	1	0,0102940	0,0064285	1,4393	0,000000
	2	0,0905100	0,1091100	0,3680	0,017027
8	1	0,0026562	0,000000	1,3200	0,000000
	2	0,0715960	0,000000	0,2772	0,023106

Para os cálculos, assumimos que: $\chi_1 = 1$ e $\chi_2 = 0$, $B_z^2 = 0$, e como critérios de convergência: 10^{-7} para o k_{eff} , 10^{-4} e 10^{-6} para as iterações interna e externas do fluxo de nêutrons, respectivamente. As malhas utilizadas foram $\Delta x = \Delta y = 0,963\text{cm}$ na região de simetria dos eixos e $\Delta x = \Delta y = 1,005\text{cm}$ em outras regiões, totalizando 34184 regiões.

4.2 Benchmark Koebke

A Figura 4.2 mostra o núcleo do benchmark Koebke com 1/4 de simetria, com as dimensões principais dos elementos combustíveis destacadas e os quatro tipos que compõem o núcleo (TRKOV e RAVNIK, 1994) . Os parâmetros nucleares são apresentados na Tabela 4.2. O Koebke possui uma configuração hipotética e simplificada de um núcleo de um reator PWR, com barras de controle (CAPOSSOLI, 2009) . E da mesma forma que no Biblis, baffle e refletor são homogeneizados em uma única região (tipo 4). Também utilizada a condição de vácuo no contorno.

Para os cálculos, assumimos que: $\chi_1 = 1$ e $\chi_2 = 0$, $B_z^2 = 0$, e como critérios de convergência: 10^{-7} para o k_{eff} , 10^{-4} e 10^{-6} para as iterações internas e externas do fluxo de nêutrons, respectivamente. As malhas utilizadas foram $\Delta x = \Delta y = 1,00\text{cm}$

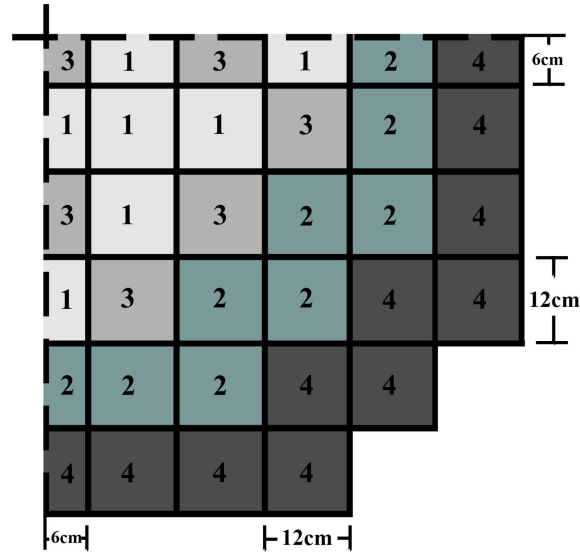


Figura 4.2: Núcleo do benchmark Koebke com 1/4 de simetria.

Tabela 4.2: Parâmetros nucleares - Koebke.

Tipo	g	$\Sigma_{ag}(cm^{-1})$	$\nu\Sigma_{fg}(cm^{-1})$	$D_g(cm)$	$\Sigma_{gg'}(cm^{-1})$
1	1	0,0080867	0,0047576	1,50400	0,00000000
	2	0,0694918	0,0857355	0,35629	0,01873042
2	1	0,0081108	0,0051284	1,52340	0,00000000
	2	0,0858192	0,1216174	0,36045	0,01799933
3	1	0,0100475	0,0046210	1,43368	0,00000000
	2	0,0789324	0,0757924	0,30619	0,01805838
4	1	0,0014103	0,0000000	1,02020	0,00000000
	2	0,1004083	0,0000000	1,21120	0,03087685

na região de simetria dos eixos e $\Delta x = \Delta y = 1,00cm$ em todas as regiões que compõem o núcleo, totalizando 3924 regiões.

4.3 Benchmark de Pesquisa de Criticalidade, Queima e Depleção

O benchmark usado para análise de 1 ciclo completo é um modelo bidimensional de um PWR de grande porte com dois ciclos de carregamento. Esta tese, no entanto, fará a análise apenas do 1º ciclo de carregamento. A documentação desse benchmark pode ser encontrada em KOEBKE *et al.* (1984) . Originalmente, foi usado para testar a capacidade de métodos de malha grossa. No entanto, nesta tese, ele será usado para cálculos em malha fina para pesquisa de criticalidade, queima e depleção para apenas 1 ciclo de carregamento. A seguir são descritos: configuração do benchmark com suas dimensões principais e a localização de elementos contendo veneno queimável; cadeia de depleção dos nuclídeos; concentração crítica de boro;

densidades iniciais dos núclídeos de urânio e veneno queimável; seções de choque microscópicas de fissão e captura dos núclídeos fissionáveis; seções de choque microscópicas de captura do veneno queimável, do xenônio, dos produtos de fissão e boro solúvel; rendimentos dos produtos de fissão e seções de choque macroscópicas do material estrutural.

A Figura 4.3 mostra núcleo do benchmark com 1/8 de simetria, com as dimensões principais dos elementos combustíveis do primeiro ciclo de carregamento destacadas, os três tipos de EC, a região do refletor, assim como os elementos contendo veneno queimável, indicados por VQ. Esse benchmark pode ser modelado tanto na simetria de 1/8 quanto na de 1/4. Esta última é a configuração usada neste trabalho, de modo a aproveitar o modo como foi modelado o gerador de núcleos (discutido na seção seguinte) e evitar o tratamento de condição de espelho, que acontece quando um núcleo é modelado na simetria de 1/8. Para isso, os respectivos elementos simétricos são adicionados. Para o tratamento do contorno também foi utilizada a condição tipo vácuo.

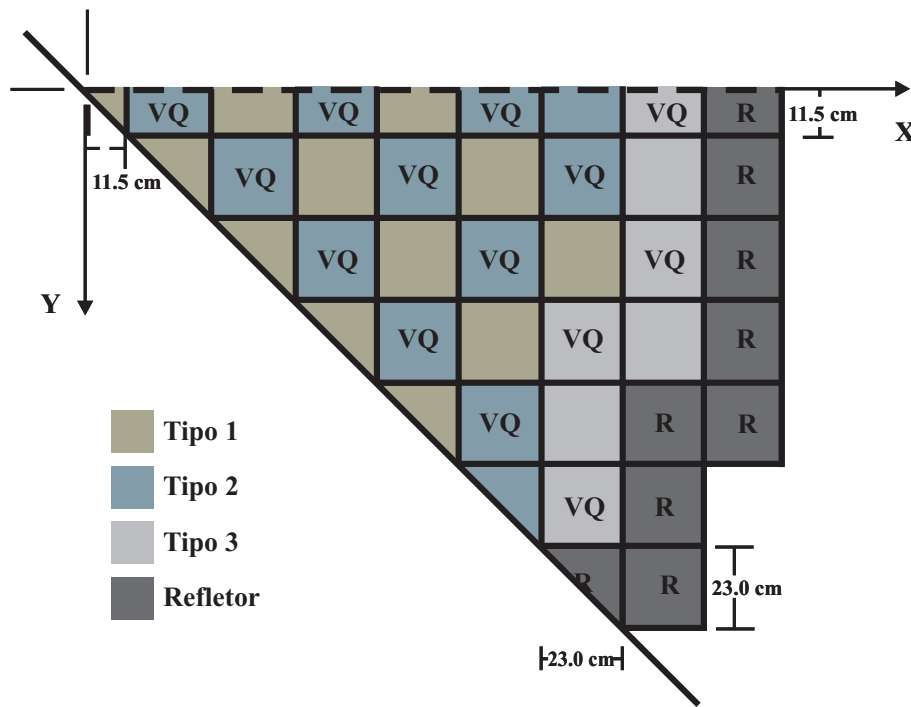


Figura 4.3: Núcleo do Benchmark 19 para o 1º ciclo de carregamento com 1/8 de simetria.

A cadeia de depleção de núclídeos é mostrada na Figura 4.4. As equações de depleção dos núclídeos (de 1 a 8), dos produtos de fissão (10) e a equação de equilíbrio do xenônio (9) encontram-se no Apêndice A.

O reator é mantido crítico durante todo ciclo ajustando-se a concentração de boro¹. A Equação 4.1 mostra como a criticalidade é função da concentração de

¹O Apêndice B detalha melhor como funciona o módulo da concentração de boro, da pesquisa

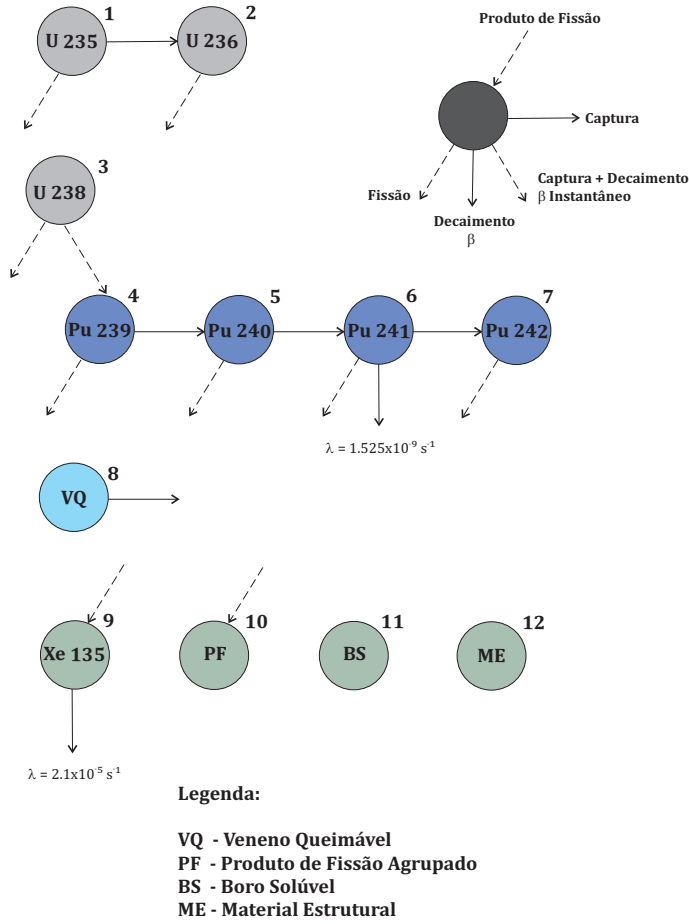


Figura 4.4: Cadeia de depleção de nuclídeos.

boro:

$$[CB]^{i+1} = \frac{(1 - k_{eff}^{(i-1)})[CB]^i - (1 - k_{eff}^{(i)})[CB]^{(i-1)}}{k_{eff}^{(i)} - k_{eff}^{(i-1)}}, \quad i = 1, 2, 3, \dots, N, \quad (4.1)$$

onde $k_{eff}^{(i)}$ e $[CB]^i$ representam o fator de multiplicação e a concentração de boro, em partes por milhão (ppm), da i -ésima iteração, respectivamente.

A concentração crítica de boro e a queima média acumulada para cada intervalo, medido em Dias Efetivos à Plena Potência (DEPP), estão relacionadas como mostra a Tabela 4.3, onde são apresentados os valores da queima média acumulada de acordo com a documentação do benchmark, e das concentrações críticas de boro para o número de dias efetivos a plena potência do reator para o 1^o ciclo de carregamento do benchmark da Figura 4.3. Os valores de queima desta tabela não dependem das seções de choque, e, conseqüentemente, nem do fluxo de nêutrons, por isso serão os mesmos que serão usados nas configurações obtidas com gerador de núcleos. No

de criticalidade. Aqui só será apresentada a equação da concentração de boro em função do fator de multiplicação efetivo, que torna $k_{eff} = 1$

Apêndice D é mostrado como esses valores são calculados, com base na potência nominal e de operação, no fator de carga, no volume do núcleo e da densidade de metal pesado.

Tabela 4.3: Concentração crítica de boro $\times$ queima média por DEPP.

Instante DEPP (dias)	Queima Média (MWD/KgU)	Concentração Crítica de Boro (ppm)
0	0	1277,8 (sem Xe)
0	0	1038,6 (Xe equilíbrio)
30	1,098	1081,7
60	2,197	1069,2
90	3,295	1019,0
120	4,394	944,1
150	5,492	853,1
180	6,590	751,5
210	7,689	643,5
240	8,787	531,7
270	9,886	418,1
300	10,984	304,3
330	12,082	191,0
360	13,181	79,1
381,49	13,968	0

As densidades iniciais dos nuclídeos de urânio e do veneno queimável² são apresentadas na Tabela 4.4. Todos os demais nuclídeos tem densidades iniciais igual a zero.

Tabela 4.4: Densidades iniciais dos nuclídeos ($10^{24} \text{atomos/cm}^3$).

Nuclídeo	Tipos		
	Tipo1	Tipo2	Tipo3
U^{235}	$1,40 \times 10^{-4}$	$1,60 \times 10^{-4}$	$2,2 \times 10^{-4}$
U^{238}	$6,31 \times 10^{-3}$	$6,29 \times 10^{-3}$	$6,23 \times 10^{-3}$
VQ	0,00	$8,00 \times 10^{-6}$	$8,00 \times 10^{-4}$

As seções de choque microscópicas dos nuclídeos fissionáveis (1 a 7) e as de captura dos nuclídeos numerados de 8 a 11³ são apresentadas nas Tabelas 4.5 e 4.6, respectivamente.

²Somente os tipos 2 e 3 contém veneno queimável, como indicado na Figura 4.3.

³A unidade para o boro (nuclídeo 11) é $\text{cm}^{-1} \text{ppm}^{-1}$.

Tabela 4.5: Seções de choque microscópicas dos núclídeos fissionáveis ($10^{-24}cm^2$).

Nuclídeo j	Grupo g	Tipos						Nêutrons por fissão ν_j^g
		Tipo1		Tipo2		Tipo3		
		σ_f^g	σ_c^g	σ_f^g	σ_c^g	σ_f^g	σ_c^g	
1 U^{235}	1	7,60	4,20	7,50	4,10	7,40	4,00	2,44
	2	238,60	42,90	231,90	41,80	220,60	39,80	2,44
2 U^{236}	1	0,20	7,90	0,20	7,80	0,20	7,70	2,79
	2	0,00	2,70	0,00	2,70	0,00	2,60	0,00
3 U^{238}	1	0,12	0,85	0,12	0,84	0,12	0,84	2,85
	2	0,00	1,23	0,00	1,20	0,00	1,15	0,00
4 Pu^{239}	1	9,80	6,40	9,70	6,40	9,70	6,40	2,90
	2	617,00	348,10	611,60	347,00	601,50	344,50	2,88
5 Pu^{240}	1	0,60	10,00	0,60	10,00	0,60	10,00	3,10
	2	0,10	915,90	0,10	977,50	0,10	1081,20	2,89
6 Pu^{241}	1	18,50	4,70	18,40	4,70	18,20	4,70	3,02
	2	635,20	222,80	622,30	218,90	599,90	212,00	2,94
7 Pu^{242}	1	0,50	36,50	0,50	36,20	0,50	35,70	3,11
	2	0,00	9,90	0,00	9,80	0,00	9,50	0,00
χ_j^g , a energia liberada por fissão é $\chi_j^g = 0,32 \times 10^{-10}W.s$ para todos os núclídeos j , por grupo g e por tipo								

Tabela 4.6: Seções de choque microscópicas de captura do veneno queimável, xenônio, produtos de fissão e do boro solúvel.

Nuclídeo j	Grupo g	$\sigma_c^g(10^{-24}cm^2)$
8 VQ	1	28,00
	2	800,00
9 Xe^{135}	1	0,00
	2	$1,20 \times 10^6$
10 PF	1	5,3
	2	36,5
11 B	1	$1,40 \times 10^{-7}$
	2	$9,38 \times 10^{-6}$

A Tabela 4.7 apresenta o valor da constante (γ_k^j), relativa aos rendimentos de xenônio e dos produtos de fissão. Essa constante é usada nas equações de equilíbrio do xenônio e de depleção dos produtos de fissão, Equações A.10 e A.11 do Apêndice A, respectivamente.

Tabela 4.7: Rendimentos do xenônio e dos produtos de fissão.

Nuclídeo	Rendimentos do xenônio e dos produtos de fissão (γ_k^j)						
$k \backslash j$	U^{235} $j = 1$	U^{236} $j = 2$	U^{238} $j = 3$	Pu^{239} $j = 4$	Pu^{240} $j = 5$	Pu^{241} $j = 6$	Pu^{242} $j = 7$
$k = 9 \text{ Xe}^{135}$	0,0663	0,0665	0,0665	0,0747	0,0665	0,0708	0,0665
$k = 10 \text{ PF}$	1,0	1,0	1,0	1,0	1,0	1,0	1,0

A Tabela 4.8 apresenta as seções de choque macroscópicas de captura, transporte⁴ e espalhamento do material estrutural.

Tabela 4.8: Seções de choque macroscópicas do material estrutural.

Tipo	Grupo	$\Sigma_c^g(cm^{-1})$	$\Sigma_{tr}(cm^{-1})$	$\Sigma_{gg'}(cm^{-1})$
Tipo1	1	$6,51 \times 10^{-4}$	0,222	$1,89 \times 10^{-2}$
	2	$7,02 \times 10^{-3}$	0,835	0,00
Tipo2	1	$6,50 \times 10^{-4}$	0,222	$1,88 \times 10^{-2}$
	2	$6,93 \times 10^{-3}$	0,829	0,00
Tipo3	1	$6,50 \times 10^{-4}$	0,222	$1,85 \times 10^{-2}$
	2	$6,78 \times 10^{-3}$	0,820	0,00
Refletor	1	$9,80 \times 10^{-4}$	0,257	$2,37 \times 10^{-2}$
	2	0,138	1,31	0,00

4.4 Gerador de Núcleos

O Modelo de Chaves Aleatórias (*Random Keys, RK*) (BEAN, 1994) , foi usado para gerar diferentes configurações do núcleo do benchmark usado nesta tese. Neste modelo, a transformação de uma fatia do reator num vetor de dimensão igual ao número de elementos desta fatia (no qual cada posição corresponde a um elemento combustível) permite manipular os elementos desse vetor para gerar uma nova configuração do núcleo. Trata-se do modelo mais conhecido de decodificação indireta usado em algoritmos metaheurísticos para otimização de recarga (LI *et al.*, 2022) .

O Modelo de Chaves Aleatórias consiste em mapear um vetor composto de números reais em uma solução contendo inteiros não repetidos. Para exemplificar o funcionamento do Modelo de Chaves Aleatórias, toma-se como exemplo o vetor $B = [2; 4; 6; 8; -1; 1; -0.5]$. Os elementos do vetor B são gerados por um gerador de números randômicos entre -10 e 10. O mapeamento do vetor B resulta no vetor auxiliar $B^* = [7; 5; 6; 1; 2; 3; 4]$.

Observando-se o vetor B, seu menor valor $B[-0.5]$ aparece na sétima posição. No vetor B^* esta posição ocupará a primeira posição. O segundo valor do vetor

⁴Para o cálculo do coeficiente de difusão por grupo - $D_g = \frac{1}{3\Sigma_{trg}}$.

$B[-1]$ aparece na quinta posição, e em B^* na segunda posição. O terceiro valor do vetor $B[1]$ aparece na sexta posição, e em B^* na terceira posição, e assim por diante, reorganizando os valores do vetor B em ordem crescente das posições ocupadas dos menores valores. Se houver dois valores repetidos, eles ocuparão posições diferentes no vetor B , e, no vetor B^* a primeira posição será considerada primeiro, isto é, ocupará a primeira posição em B^* , gerando sempre um vetor B^* sem repetições.

Usando o Biblis como exemplo, um gerador de números randômicos gera um vetor de 31 dimensões conforme mostra a Figura 4.5a. Outro vetor codifica as posições como chaves, como mostra a Figura 4.5b, e o vetor auxiliar decodifica o vetor de posições, rearranjando os valores reais gerados em ordem crescente, como mostra a Figura 4.5c.

(a)	2	4	6	8	-1	1	-0.5	-3	-0.1	-4	5	-7.2
	1.1	3	5.1	7	2.9	9	10	5.8	-0.6	8.3	9.7	6.8
	-9.5	-8	-7	-6.5	-6	-4.7	-3.4					

(b)	1	2	3	4	5	6	7	8	9	10	11	12
	13	14	15	16	17	18	19	20	21	22	23	24
	25	26	27	28	29	30	31					

(c)	25	26	12	27	28	29	30	10	31	8	5	21
	7	9	6	13	1	17	14	2	11	15	20	3
	24	16	4	22	18	23	19					

Figura 4.5: Esquema de codificação em chaves aleatórias de um vetor com 31 dimensões.

A Figura 4.6d mostra o vetor de 31 dimensões correspondente aos elementos de octeto, o elemento central e os de quarteto dos eixos principal e da diagonal, excluindo o tipo 8 (baffle + refletor) do Biblis. Depois de decodificado, as posições do vetor auxiliar 4.5c ocupam as novas posições de acordo com a configuração do vetor original Figura 4.6e.

Os tipos das posições originais, também se movem para as respectivas posições do vetor auxiliar, como vemos na Figura 4.7. A reconstrução da simetria de 1/4 é feita rebatendo-se os elementos de octeto, isto é, incluindo os simétricos, e depois, completando a montagem completa do núcleo com o tipo 8 (baffle + refletor), fixos

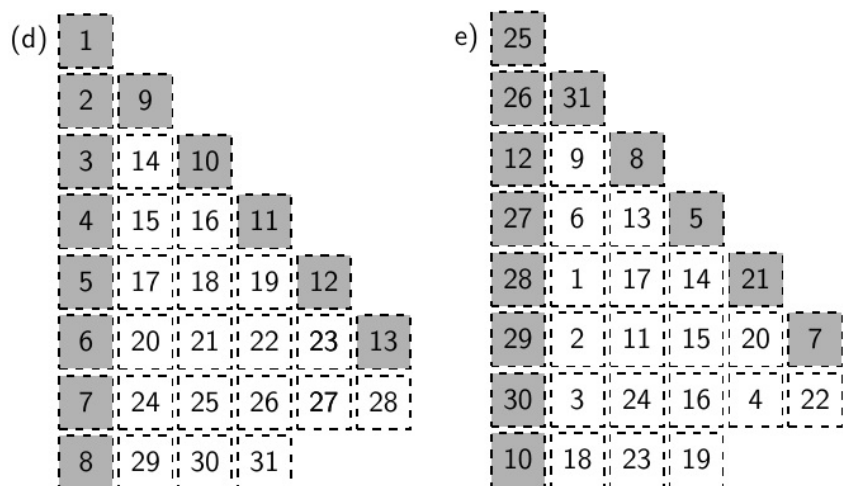


Figura 4.6: Núcleo resultante da decodificação em chaves aleatórias.

na fronteira do sistema, como mostra a Figura 4.8. A reconstrução da simetria de $1/4$ é necessária porque o benchmark e suas configurações serão simulados desta forma.

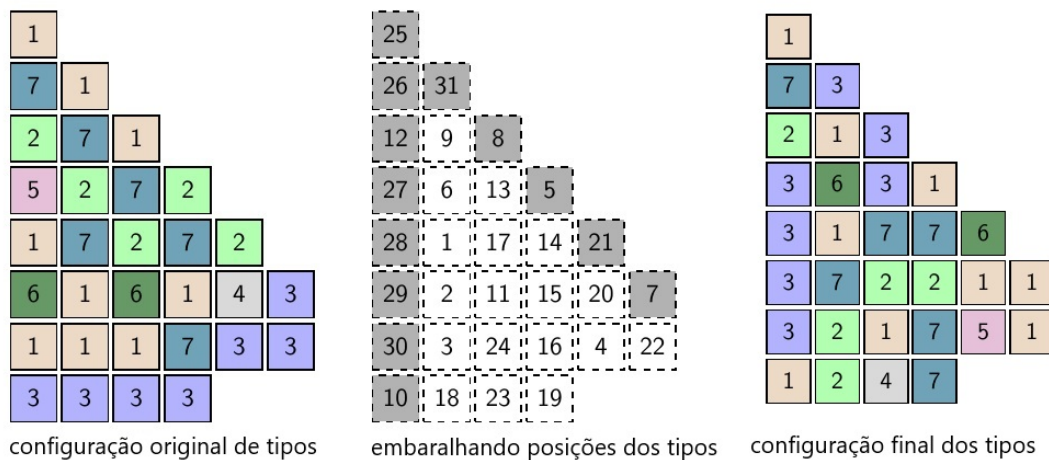
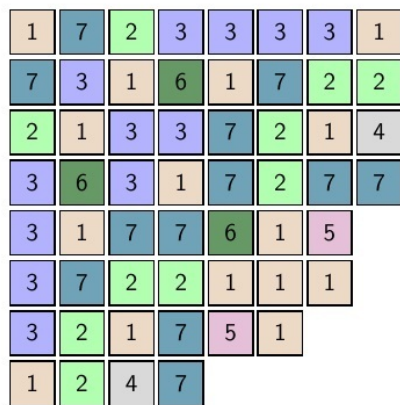
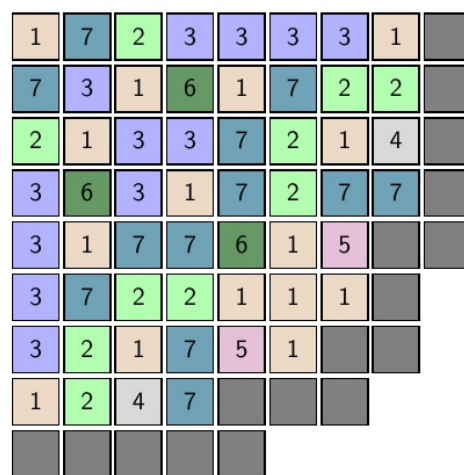


Figura 4.7: Núcleo resultante com os tipos em novas posições.



Inserindo os tipos simétricos



Completando com o tipo 8

Figura 4.8: Núcleo resultante final incluindo tipos simétricos e tipo 8.

Capítulo 5

Resultados

Neste capítulo são apresentados os resultados numéricos utilizando as duas teorias: 2 grupos e 1 grupo modificado de energia. Os resultados relacionados às otimizações no compilador Intel FORTRAN aplicadas a teoria de 1 grupo modificado são também analisados.

Este capítulo é dividido em duas seções: uma de resultados relativos à validação, e outra de análise 1 ciclo completo de queima. A primeira parte é relativa à aplicabilidade da teoria de 1 grupo modificado de energia para determinação do k_{eff} e distribuição de potência, mostrando os desvios em relação à teoria de 2 grupos usando 2 benchmarks diferentes, um reator de grande porte, o Biblis e um outro de pequeno porte, o Koebke. Este último, conforme já afirmado na introdução do capítulo anterior, foi usado para avaliar se a teoria de 1 grupo modificado, mesmo sendo restrita a reatores de grande porte, produziria resultados dentro da mesma margem de erro quando aplicada a reatores de grande porte. Na segunda parte é feita a análise de 1 ciclo completo de queima, usando outro benchmark de grande porte, o *PWR-Burnup Problem - with fuel management (Benchmark 19)*, cujas concentrações iniciais dos núclídeos são conhecidas, realizando pesquisa de criticalidade, queima e depleção para as duas teorias e com 1 grupo modificado otimizado para o 1º ciclo de carregamento (*Cycle 1*). Vale salientar que nesta segunda parte, a diferença entre as duas teorias é apenas no cálculo do fluxo. Como consequência, os cálculos de outras grandezas que usam o fluxo diferem entre si quando usamos duas teorias diferentes, mas sem a necessidade de alterar a programação para a pesquisa de criticalidade, queima e depleção.

A solução numérica foi da mesma forma que a desenvolvida por (DA SILVA *et al.*, 2023) e implementadas em Intel FORTRAN. As simulações foram realizadas em um computador com as seguintes configurações: PC com processador Intel Core i7-4790K de 4 GHz, placa-mãe Z97X-UD5H-BK com 16 GB RAM (DDR3 1600 MHz) instalada com sistema operacional Windows 10 Pro x64. A paralelização foi feita nos *check boxes* do menu de propriedades do projeto do Visual Studio, como

mostrado no Apêndice C.

5.1 Biblis

Os resultados para o Biblis (Figura 4.2) para o cálculo do k_{eff} estão na Tabela 5.1. Nela são comparados os resultados do tempo computacional total para simulações usando dois grupos de energia e um grupo modificado sem e com otimização, mostrando os valores do k_{eff} e do erro relativo¹, além do fator de multiplicação de referência MÜLLER e WEISS (1991) . As otimizações no compilador Intel FORTRAN aplicadas à teoria de 1 grupo modificado reduzem significativamente o tempo de processamento total, sem alteração dos resultados, afetando apenas o tempo computacional.

Tabela 5.1: Resultados para o tempo de processamento do Biblis.

Teoria	k_{eff}	Erro Relativo (%)	Tempo (s)
2 grupos	1,025255	0,0142	172,76
1 grupo modificado	1,026416	0,1274	64,95
1 grupo modificado (otimizado)	1,026416	0,1274	13,70

$$k_{eff_{REFERENCIA}} = 1,025110$$

A Tabela 5.2 mostra o cálculo da distribuição de potência normalizada (DA SILVA, 2011) e o erro relativo porcentual² usando as duas teorias para o Biblis, destacando os erros relativos porcentuais máximos para cada uma das teorias: 4,80% para 1 grupo modificado, e 0,50% para 2 grupos em relação à referência MÜLLER e WEISS (1991) .

Usando o gerador, 100 configurações diferentes do núcleo do reator Biblis foram geradas. O resultado do k_{eff} para as 100 configurações com as duas teorias é mostrado na Figura 5.1 corroborando os desvios do k_{eff} da Tabela 5.1 em relação à teoria de 2 grupos. A Figura 5.2 mostra que o desvio médio do k_{eff} para as 100 configurações ficou em torno de 0,1%. O tempo computacional para cada configuração, para as duas teorias, e para 1 grupo modificado otimizado são mostrados no gráfico da Figura 5.3, mostrando que o uso da teoria de 1 grupo modificado sem otimização reduziu para mais de 50% o tempo de processamento, e com otimizações, para mais de 80%.

¹Erro relativo do k_{eff} : $[(k_{eff_{REF}} - k_{eff})/k_{eff_{REF}}] \times 100\%$.

²Erro relativo: $[(\bar{P}_{EC(REF)}^N - \bar{P}_{EC}^N)/\bar{P}_{EC(REF)}^N] \times 100\%$.

Tabela 5.2: Distribuição de potência normalizada do Biblis.

1,0911	1,1016	1,2428	1,2205	1,0888	0,9820	1,0947	1,0146
3,61%	3,10%	3,16%	1,32%	3,78%	2,81%	3,03%	1,80%
0,11%	0,33%	0,22%	0,23%	0,16%	0,21%	0,50%	0,26%
1,1016	1,1176	1,1341	1,2237	1,0677	1,0319	1,0718	0,9705
3,10%	3,83%	2,69%	3,24%	2,72%	3,05%	2,45%	1,95%
0,33%	0,11%	0,31%	0,13%	0,07%	0,37%	0,45%	0,25%
1,2428	1,1341	1,1223	1,1051	1,1200	0,9236	0,9309	0,8245
3,16%	2,69%	4,05%	2,45%	3,07%	2,84%	2,89%	2,55%
0,22%	0,31%	0,06%	0,25%	0,00%	0,07%	0,37%	0,15%
1,2205	1,2237	1,1051	1,1608	1,0387	0,9501	0,7651	0,5454
1,32%	3,24%	2,45%	3,31%	2,63%	3,17%	3,36%	4,80%
0,23%	0,13%	0,25%	0,09%	0,17%	0,13%	0,01%	0,02%
1,0888	1,0677	1,1200	1,0387	1,1225	0,9929	0,8743	
3,78%	2,72%	3,07%	2,63%	2,69%	0,36%	2,48%	
0,16%	0,07%	0,00%	0,17%	0,08%	0,14%	0,17%	
0,9820	1,0319	0,9236	0,9501	0,9929	1,1995	0,6842	
2,81%	3,05%	2,84%	3,17%	0,36%	0,07%	3,33%	
0,21%	0,37%	0,07%	0,13%	0,14%	0,22%	0,31%	
1,0947	1,0718	0,9309	0,7651	0,8743	0,6842		
3,03%	2,45%	2,89%	3,36%	2,48%	3,33%		
0,50%	0,45%	0,37%	0,01%	0,17%	0,31%		
1,10146	0,9705	0,8245	0,5454				Referência
1,80%	1,95%	2,55%	4,80%				1 Grupo Modificado
0,26%	0,25%	0,15%	0,02%				2 Grupos

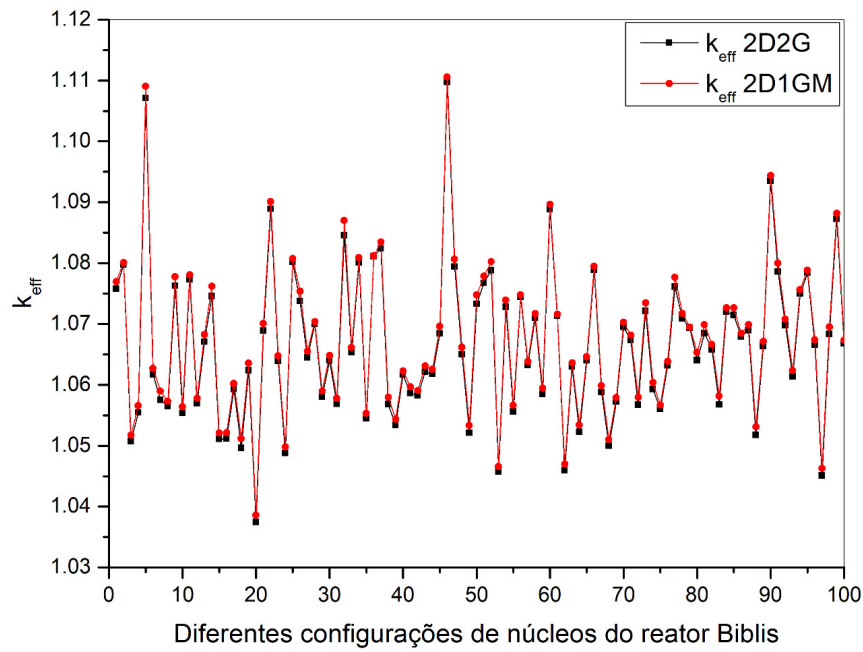


Figura 5.1: Valor do k_{eff} para 100 configurações do Biblis.

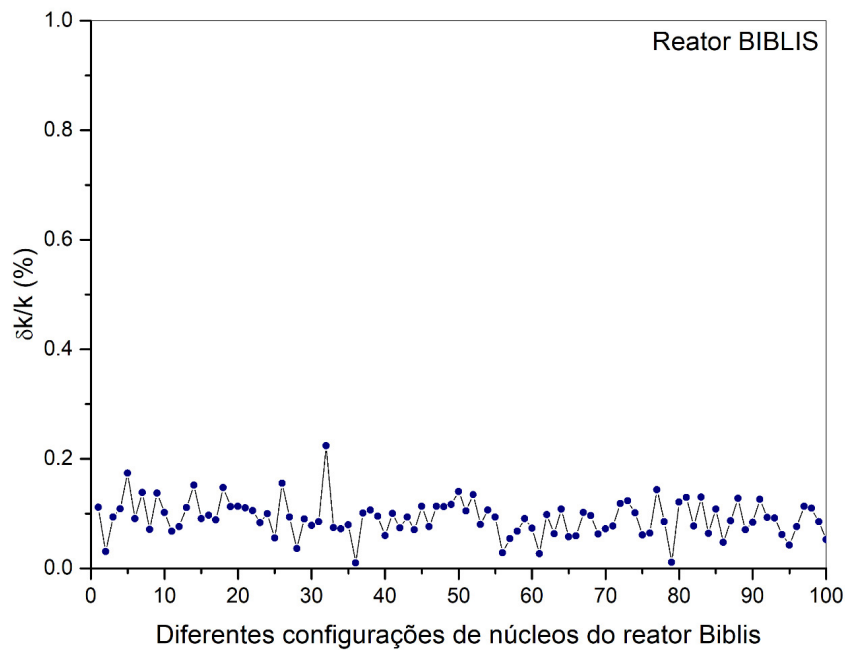


Figura 5.2: Desvio do k_{eff} para 100 configurações do Biblis.

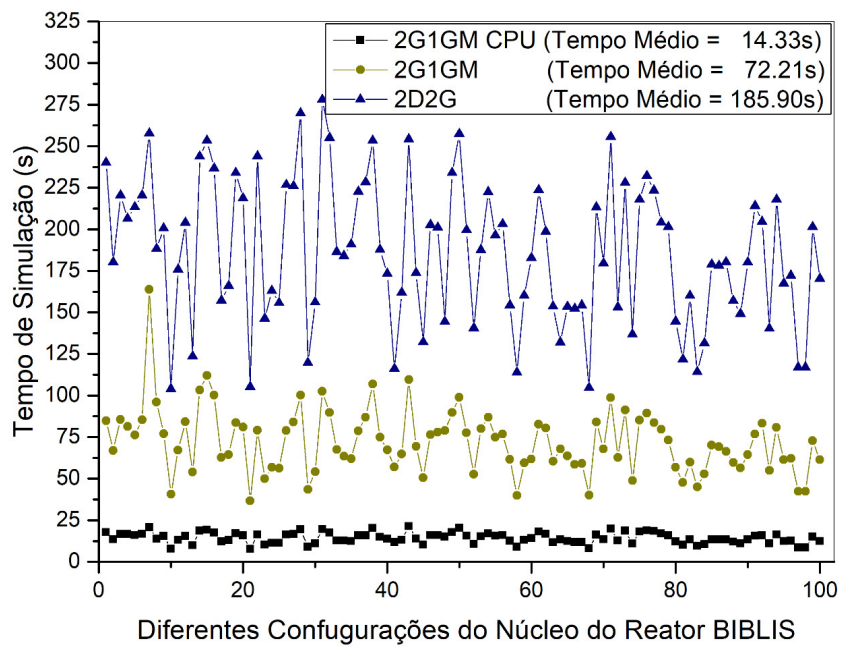


Figura 5.3: Tempo médio de processamento para 100 configurações do Biblis.

5.2 Koebke

Da mesma forma, os resultados do k_{eff} para o Koebke (Figura 4.2) estão na Tabela 5.3. Nela são comparados os resultados do tempo computacional total para simulações usando dois grupos de energia e um grupo modificado sem e com otimização, mostrando os valores do k_{eff} e do erro relativo, além do fator de multiplicação de referência (TRKOV e RAVNIK, 1994) .

Tabela 5.3: Resultados para o tempo de processamento do Koebke.

Teoria	k_{eff}	Erro Relativo (%)	Tempo (s)
2 grupos	0,902376	0,0348	3,22
1 grupo modificado	0,909405	0,7439	1,09
1 grupo modificado (otimizado)	0,909405	0,7439	0,17

$$k_{eff_{REFERENCIA}} = 0,90269$$

A Tabela 5.4 mostra o cálculo da distribuição de potência normalizada (DA SILVA, 2011) e o erro relativo porcentual usando as duas teorias para o Biblis, destacando os erros relativos porcentuais máximos para cada uma das teorias em relação à referência (TRKOV e RAVNIK, 1994) . Como se pode observar, o maior erro relativo porcentual no cálculo da distribuição de potência normalizada foi de 5,04% para a teoria de um grupo modificado e de 0,84% para a de dois grupos. De acordo com CAPOSSOLI (2009) , como o Koebke é um benchmark com barras de controle, os erros maiores que 5% estão nas regiões mais internas do núcleo devido a presença das barras de controle, apresentando uma diminuição do fluxo de nêutrons devido à absorção, e, conseqüentemente, um erro na potência calculada para estas regiões. Este valor está dentro da margem aceitável para a teoria de 1 grupo modificado e está próximo do valor obtido para o Biblis, que é um reator de grande porte.

Novamente, usando o gerador 100 configurações diferentes do núcleo do Koebke foram geradas. O resultado do k_{eff} para as 100 configurações com as duas teorias é mostrado na Figura 5.4, corroborando os desvios do k_{eff} da Tabela 5.3. A Figura 5.5 mostra que o desvio médio do k_{eff} para as 100 configurações ficou em torno de 0,5%. O tempo médio computacional para cada configuração, para as duas teorias, e para 1 grupo modificado otimizado são mostrados no gráfico da Figura 5.6, mostrando o mesmo efeito de redução de tempo computacional que foi observado para o Biblis (ver Figura 5.3).

Tabela 5.4: Distribuição de potência normalizada do Koebke.

1,356 5,01% 0,087%	1,601 2,06% 0,25%	1,103 5,04% 0,11%	1,030 4,68% 0,10%	0,830 1,75% 0,62%
1,601 2,06% 0,25%	1,575 0,92% 0,51%	1,299 2,69% 0,13%	0,811 1,60% 0,84%	0,769 0,45% 0,33%
1,103 5,04% 0,11%	1,299 2,69% 0,13%	0,921 2,06% 0,72%	1,059 1,00% 0,27%	0,618 0,34% 0,28%
1,030 4,68% 0,10%	0,811 1,60% 0,84%	1,059 1,00% 0,27%	0,742 0,56% 0,08%	
0,830 1,75% 0,62%	0,769 0,45% 0,33%	0,618 0,34% 0,28%	Referência 1 Grupo Modificado 2 Grupos	

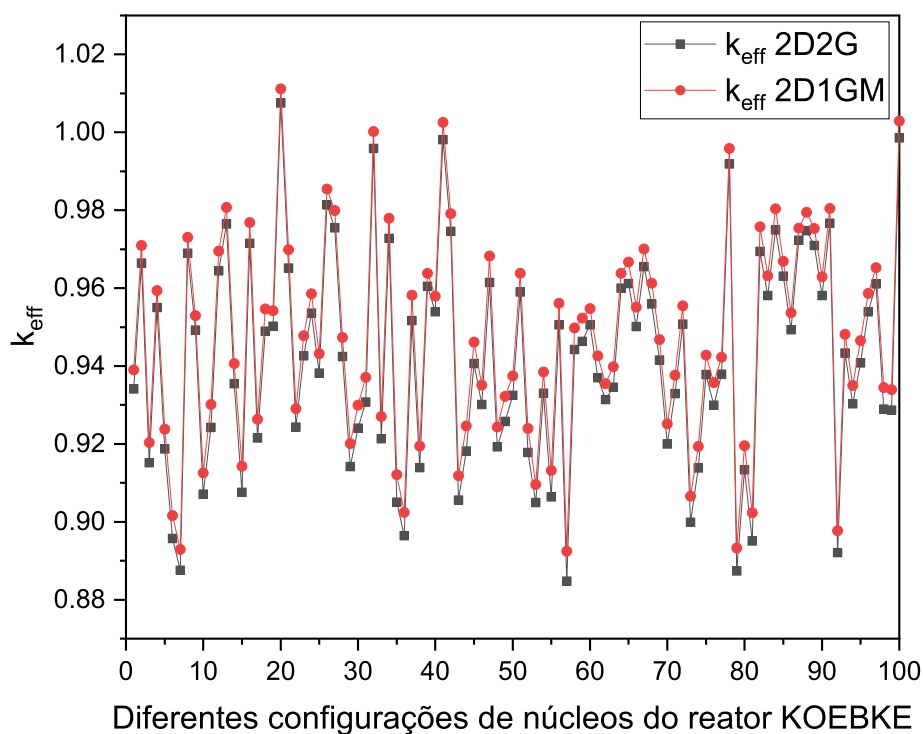


Figura 5.4: Valor do k_{eff} para 100 configurações do Koebke.

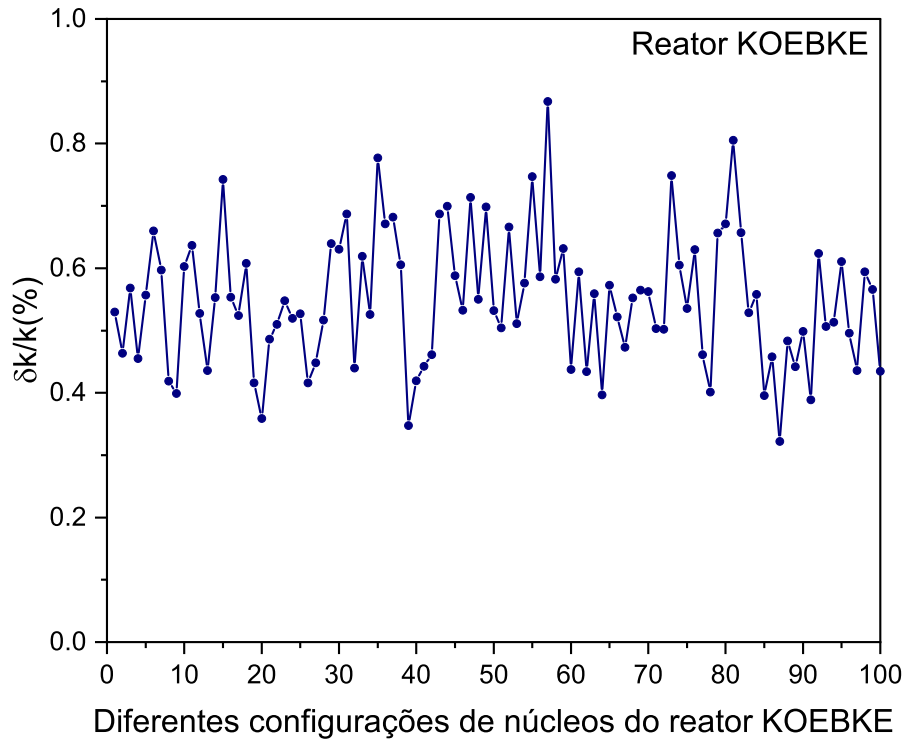


Figura 5.5: Desvio do k_{eff} para 100 configurações do Koebke.

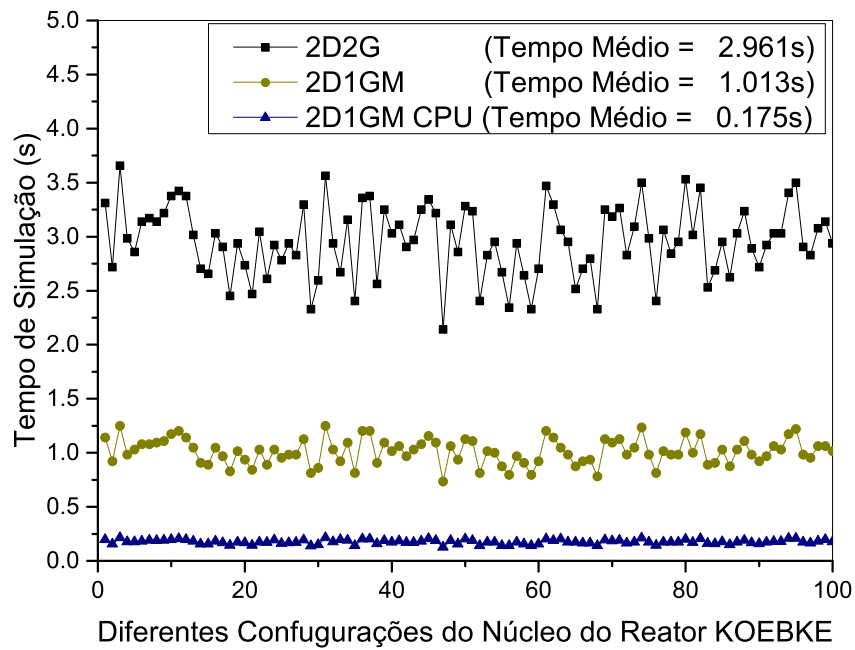


Figura 5.6: Tempo médio de processamento para 100 configurações do Koebke.

5.3 Análise de 1 ciclo completo

5.3.1 Benchmark 19

A Tabela 5.5 mostra os resultados da distribuição de potência normalizada (DA SILVA, 2011) para a configuração da Figura 4.3 referente ao Benchmark 19 (**Two-dimensional PWR-Burnup Problem - with Fuel Management**) no início do 1º ciclo (*BOC1 - Beginning of Cycle 1*) antes do equilíbrio de xenônio. São comparados os resultados para as duas teorias em relação à referência KOEBKE *et al.* (1984).

Tabela 5.5: Distribuição de potência normalizada em BOC1 sem xenônio.

1,114 8,08% -0,45%	1,128 12,5% -0,35%	1,107 6,50% -0,45%	1,116 9,59% -0,27%	1,092 1,74% -0,27%	1,106 2,98% -0,09%	1,137 -6,33% -0,09%	0,863 -4,75% -0,46%
1,128 12,5% -0,35%	1,110 7,30% -0,45%	1,121 11,15% -0,27%	1,096 4,20% -0,36%	1,103 6,62% -0,18%	1,074 -2,33% -0,19%	1,036 -2,22% 0,10%	0,896 -8,03% 0,45%
1,107 6,50% -0,45%	1,121 11,15% -0,27%	1,097 5,01% -0,36%	1,103 8,25% -0,18%	1,075 0,37% -0,28%	1,080 2,04% 0,00%	0,996 -8,93% 0,00%	0,782 -5,24% 0,51%
1,116 9,59% -0,27%	1,096 4,20% -0,36%	1,103 8,25% -0,18%	1,070 1,31% -0,28%	1,068 3,93% -0,09%	1,050 -5,90% 0,00%	1,158 -2,94% 0,34%	0,639 -11,27% 0,78%
1,092 1,74% -0,27%	1,103 6,62% -0,18%	1,075 0,37% -0,28%	1,068 3,93% -0,09%	1,009 -3,86% 0,00%	0,964 -2,90% 0,21%	0,916 -8,84% 0,44%	Referência 1 grupo Modificado 2 Grupos
1,106 2,98% 0,09%	1,074 -2,33% -0,19%	1,080 2,04% 0,00%	1,050 -5,90% 0,00%	0,964 -2,90% 0,21%	0,816 -9,68% 0,24%	0,503 -6,36% 0,79%	
1,137 -6,33% -0,09%	1,036 -2,22% 0,10%	0,996 -8,93% 0,00%	1,158 -2,94% 0,34%	0,916 -8,84% 0,44%	0,503 -6,36% 0,79%		
0,863 -4,75% 0,46%	0,896 -8,03% 0,45%	0,782 -5,24% 0,51%	0,639 -11,27% 0,78%				

A Tabela 5.6 mostra os resultados da distribuição de potência normalizada para a configuração da Figura 4.3 no final do 1º ciclo (*EOC1 - End of Cycle 1*). São comparados os resultados para as duas teorias em relação à referência. Comparando-a com a Tabela 5.5, a teoria de 1 grupo modificado apresenta resultados mais precisos depois do equilíbrio de xenônio, onde o maior desvio em relação à referência é 3,86%, na fronteira com o refletor.

Tabela 5.6: Distribuição de potência normalizada em EOC1.

1,036	1,083	1,038	1,089	1,046	1,095	1,057	0,868
-0,19%	0,65%	-0,10%	0,92%	0,29%	1,19%	-0,09%	-0,80%
0,19%	0,18%	0,19%	0,27%	0,09%	0,09%	-0,09%	-0,23%
1,083	1,037	1,086	1,042	1,094	1,045	1,052	0,872
0,65%	-0,10%	0,83%	0,09%	1,19%	0,29%	0,19%	-1,38%
0,18%	0,19%	0,18%	0,09%	0,18%	0,00%	-0,09%	-0,23%
1,038	1,086	1,041	1,093	1,050	1,095	0,996	0,833
-0,10%	0,83%	0,00%	1,10%	0,38%	1,28%	-1,10%	-1,08%
0,19%	0,18%	0,10%	0,18%	0,09%	0,09%	-0,10%	-0,24%
1,089	1,042	1,093	1,050	1,101	1,044	1,117	0,673
0,92%	0,09%	1,10%	0,28%	1,36%	-0,19%	0,89%	-3,86%
0,27%	0,09%	0,18%	0,09%	0,18%	-0,09%	-0,09%	-0,59%
1,046	1,094	1,050	1,101	1,047	1,055	0,923	
0,29%	1,19%	0,38%	1,36%	0,29%	0,00%	-1,30%	
0,09%	0,18%	0,09%	0,18%	0,00%	-0,09%	-0,32%	
1,195	1,045	1,095	1,044	1,055	0,927	0,634	
1,19%	0,29%	1,28%	-0,19%	0,00%	-1,94%	-2,52%	
0,09%	0,00%	0,09%	-0,09%	-0,09%	-0,21%	-0,31%	
1,057	1,052	0,996	1,117	0,923	0,634		
-0,09%	0,19%	-1,10%	0,89%	-1,30%	-2,52%		
-0,09%	-0,09%	-0,10%	-0,09%	-0,32%	-0,31%		
0,868	0,872	0,833	0,673				Referência
-0,80%	-1,38%	-1,08%	-3,86%				1 Grupo Modificado
-0,23%	-0,23%	-0,24%	-0,59%				2 Grupos

A Tabela 5.7 apresenta os resultados de queima média por EC³, em MWD/kgU, para as duas teorias em relação à referência no final do 1º ciclo de carregamento (EOC1 - End of Cycle 1), onde o maior desvio⁴ da teoria de 1 grupo modificado em relação à referência foi de 5,17%, também na fronteira com o refletor.

A Tabela 5.8 mostra os valores calculados da concentração crítica de boro (CB) para as duas teorias, para cada instante de queima, e os respectivos valores de referência, em Dias Efetivos à Plena Potência (DEPP), com as concentrações de boro para o instante 0 de queima, antes e depois do equilíbrio do xenônio, indicadas por **sem Xe e Xe equilíbrio**, respectivamente. Os valores calculados com a teoria de 2 grupos são indicados por 2D2G, e com 1 grupo modificado, por 2D1GM. Baseado nesta tabela, foi determinada a curva de boro para o benchmark, Figura 5.7, usando as duas teorias (indicadas pelos ponteiros), sendo que a curva cheia é a da referência.

A Tabela 5.9 apresenta os desvios das duas teorias em relação à referência para as concentrações médias dos nuclídeos, incluindo produtos de fissão (PF) e veneno

³No Apêndice D é apresentado o cálculo de queima média em cada elemento combustível e no núcleo.

⁴Erro relativo para a queima média: $[(B_{EC(REF)} - B_{EC})/B_{EC(REF)}] \times 100$.

Tabela 5.7: Queima Média por EC em EOC1.

15,181	15,757	15,189	15,753	15,125	15,531	15,011	11,641
0,47%	2,77%	0,48%	2,74%	0,24%	1,80%	-1,62%	-1,28%
-0,06%	0,01%	-0,05%	0,01%	-0,05%	0,00%	-0,10%	0,04%
15,757	15,186	15,762	15,175	15,674	14,869	14,483	11,838
2,77%	0,48%	2,77%	0,43%	2,52%	-0,40%	-0,13%	-2,70%
0,01%	-0,06%	0,01%	-0,05%	0,01%	-0,06%	-0,02%	-0,01%
15,189	15,762	15,184	15,725	15,058	15,354	13,704	10,866
0,48%	2,77%	0,47%	2,68%	0,12%	1,74%	-2,97%	-1,46%
-0,05%	0,01%	-0,04%	0,02%	-0,04%	0,02%	-0,07%	0,08%
15,753	15,175	15,725	15,081	15,472	14,616	15,503	8,764
2,74%	0,43%	2,68%	0,23%	2,17%	-1,57%	0,33%	-5,17%
0,01%	-0,05%	0,02%	-0,04%	0,03%	-0,05%	0,00%	0,10%
15,125	15,674	15,058	15,472	14,506	14,133	12,438	
0,24%	2,52%	0,12%	2,17%	-0,76%	-0,35%	-2,84%	
-0,05%	0,01%	-0,04%	0,03%	-0,02%	0,03%	0,05%	
15,531	14,869	15,354	14,616	14,133	11,984	7,665	
1,80%	-0,40%	1,74%	-1,57%	-0,35%	-3,68%	-2,62%	
0,00%	-0,06%	0,02%	-0,05%	0,03%	0,01%	0,23%	
15,011	14,483	13,704	15,503	12,438	7,665		
-1,62%	-0,13%	-2,97%	0,33%	-2,84%	-2,62%		
-0,10%	-0,02%	-0,07%	0,00%	0,05%	0,23%		
11,641	11,838	10,866	8,764				Referência
-1,28%	-2,70%	-1,46%	-5,17%				1 Grupo Modificado
0,04%	-0,01%	0,08%	0,10%				2 Grupos

queimável (VQ), em EOC1.

Os tempos computacionais para as duas teorias e 1 grupo modificado otimizado estão na Tabela 5.10 e mostram o mesmo comportamento da redução do tempo computacional das Tabelas 5.1 e 5.3.

Tabela 5.8: Concentração crítica de boro $\times$ queima média por DEPP.

Instante DEPP (dias)	Queima Média (MWD/KgU)	CB-Calculado 2D2G (ppm)	CB-Calculado 2D1GM (ppm)	CB Referência (ppm)
0	0	1276,5	1297,1	1277,8 (sem Xe)
0	0	1040,2	1059,8	1038,6 (Xe equilíbrio)
30	1,098	1083,1	1097,9	1081,7
60	2,197	1070,2	1083,6	1069,2
90	3,295	1019,6	1033,6	1019
120	4,394	944,3	958,5	944,1
150	5,492	853	868,2	853,1
180	6,590	751,4	767,4	751,5
210	7,689	643,3	656,4	643,5
240	8,787	531,5	548,2	531,7
270	9,886	418,1	435,5	418,1
300	10,984	304,3	321,6	304,3
330	12,083	191,3	209,1	191
360	13,181	79,6	98,1	79,1
381,49	13,968	0,6	19,3	0

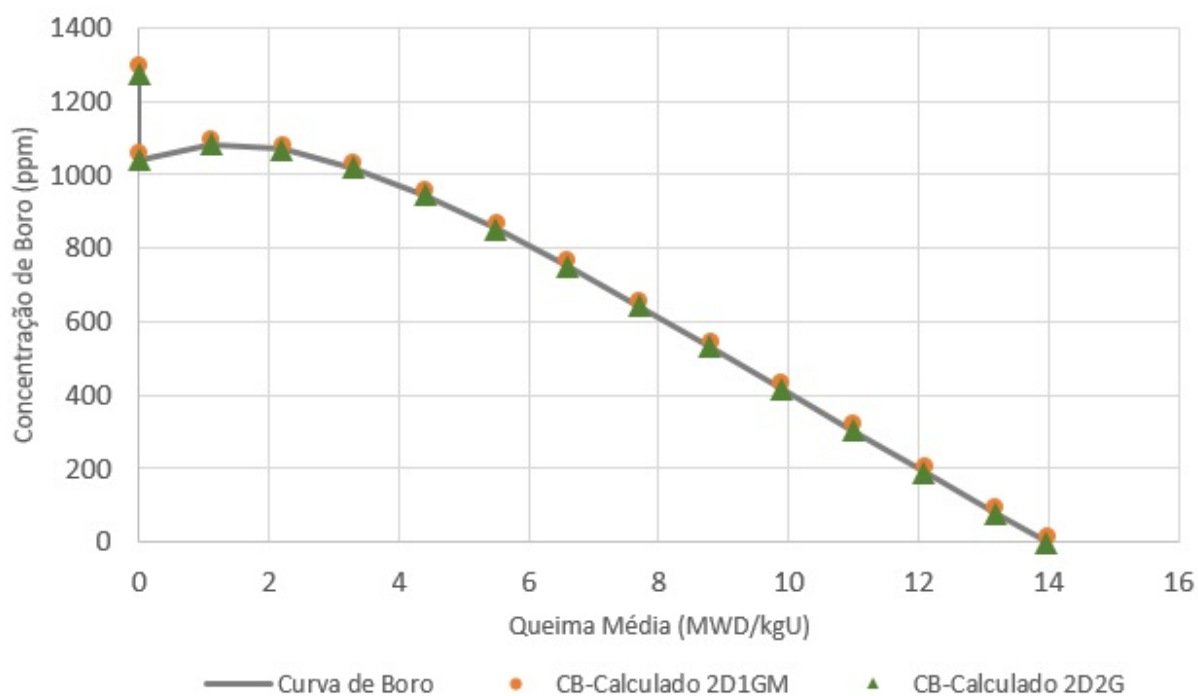


Figura 5.7: Curva de Boro.

Tabela 5.9: Concentrações Médias em EOC1.

Nuclídeo	2D2G Desvio (%)	2D1GM Desvio (%)	Concentração Referência ($10^{24} \text{ atomos/cm}^3$)
U^{235}	0,04	-0,11	$0,91245 \times 10^{-4}$
U^{236}	-0,04	0,17	$0,13751 \times 10^{-4}$
U^{238}	0,00	0,00	$0,62125 \times 10^{-2}$
Pu^{239}	0,02	0,21	$0,62125 \times 10^{-2}$
Pu^{240}	0,02	0,11	$0,65095 \times 10^{-5}$
Pu^{241}	0,01	-0,29	$0,38661 \times 10^{-5}$
Pu^{242}	-0,05	-1,28	$0,66283 \times 10^{-6}$
Xe^{135}	0,44	0,57	$0,20533 \times 10^{-8}$
PF	-0,03	-0,03	$0,95790 \times 10^{-4}$
VQ	0,24	1,45	$0,71545 \times 10^{-6}$

Tabela 5.10: Resultados para o tempo de processamento.

Teoria	Tempo (min)
2 grupos	58
1 grupo modificado	29
1 grupo modificado (otimizado)	8

5.3.2 Resultados para outras configurações

Nesta subseção são apresentados os resultados para 100 configurações diferentes do núcleo do Benchmark19 (Figura 4.3) usando o gerador de núcleos descrito no Capítulo 4. Este número de foi o mesmo adotado anteriormente, onde examinamos outras configurações dos reatores Biblis e Koebke.

Os resultados da teoria de 1 grupo modificado são comparados com aqueles obtidos usando 2 grupos de energia, tomados como referência⁵. São apresentados os resultados para as configurações cuja concentração crítica de boro ao final do ciclo foi menor que 10ppm, pois, na prática, quando essa concentração chega a valores próximos de zero⁶, pode-se considerar o fim do ciclo de operação para o carregamento em questão, e os resultados com a maior concentração de boro ao final do ciclo. Por último, são apresentados os resultados com todas as configurações e o resultado do tempo médio de simulação para as duas teorias e 1 grupo modificado otimizado. A apresentação dos resultados é feita da mesma maneira para o benchmark da configuração original, como apresentados na subseção anterior, tomando o resultado com 2 grupos de energia como a referência e para a teoria de 1 grupo modificado apenas seu desvio.

A seguir são apresentados os resultados da distribuição de potência normalizada em BOC1 antes do equilíbrio de xenônio; em EOC1; para a queima média por EC, para a concentração de boro para cada instante de queima; curva de boro; concentrações médias em EOC1; e o tempo médio de processamento para as duas teorias e com 1 grupo modificado otimizado.

I) Configurações com menor e maior concentração final de boro

A configuração com menor concentração de boro em EOC1 foi o Núcleo 93, com 1,75ppm⁷, e a configuração com maior concentração de boro em EOC1 foi o Núcleo 70, com 129ppm. Do ponto de vista do comprimento do ciclo, como podemos observar, o núcleo 93 possui um ciclo de operação (em DEPP) muito próximo do Benchmark 19, pois a sua concentração de boro está muito próxima de zero. Já o núcleo 70 poderá operar por um intervalo de tempo maior, pois ainda possui uma quantidade razoável de de boro solúvel em ppm.

A distribuição de potência normalizada para os Núcleos 93 e 70 em BOC1, antes do equilíbrio de xenônio, estão nas Tabelas 5.11 e 5.12. E nas Tabelas 5.13 e 5.14, a distribuição de potência em EOC1. Repetindo o mesmo padrão, a teoria de 1 grupo

⁵Erro relativo para a potência normalizada: $[(\bar{P}_{EC(2D2G)}^N - \bar{P}_{EC(2D1GM)}^N)/\bar{P}_{EC(2D2G)}^N] \times 100$.

⁶O Benchmark 19 adotou a concentração de 0 ppm como condição de fim de ciclo.

⁷Valor calculado com 2 grupos de energia, usado como referência.

modificado apresenta resultados mais precisos depois do equilíbrio de xenônio e os maiores desvios estão localizados na fronteira com o refletor, como destacado nas Tabelas 5.13 e 5.14 em EOC1.

Tabela 5.11: Distribuição de potência normalizada em BOC1 antes do equilíbrio do xenônio do Núcleo 93.

2,138	1,678	1,728	1,881	1,343	0,927	0,619	0,384
2,267	1,631	1,743	1,785	1,361	0,898	0,634	0,370
-6,03%	2,80%	-0,87%	5,10%	-1,34%	3,13%	-2,42%	3,64%
1,678	1,607	1,616	1,451	1,241	0,972	0,687	0,317
1,631	1,625	1,575	1,483	1,207	1,001	0,679	0,332
2,80%	-1,12%	2,54%	-2,20%	2,74%	-2,98%	1,16%	-4,73%
1,728	1,616	2,110	1,437	1,349	1,314	0,970	0,322
1,743	1,575	2,073	1,407	1,368	1,275	0,984	0,328
-0,87%	2,54%	1,75%	2,09%	-1,41%	2,97%	-1,44%	-1,86%
1,881	1,451	1,437	1,331	1,228	0,951	0,582	0,213
1,785	1,483	1,407	1,349	1,254	0,981	0,584	0,227
5,10%	-2,20%	2,09%	-1,35%	-2,12%	-3,15%	-0,34%	-6,57%
1,343	1,241	1,349	1,228	1,479	0,785	0,370	
1,361	1,207	1,368	1,254	1,451	0,810	0,384	
-1,34%	2,74%	-1,41%	-2,12%	1,89%	-3,18%	-3,78%	
0,927	0,972	1,314	0,951	0,785	0,559	0,208	
0,898	1,001	1,275	0,981	0,810	0,568	0,219	
3,13%	-2,98%	2,97%	-3,15%	-3,18%	-1,61%	-5,28%	
0,619	0,687	0,970	0,582	0,370	0,208		
0,634	0,679	0,984	0,584	0,384	0,219		
-2,42%	1,16%	-1,44%	-0,34%	-3,78%	-5,28%		
0,384	0,317	0,322	0,213				2D2G
0,370	0,332	0,328	0,227				2D1GM
3,64%	-4,73%	-1,86%	-6,57%				Desvio

Nas Tabelas 5.15 e 5.16 estão os resultados da teoria de 1 grupo modificado para queima média por EC⁸, em MWD/kgU, em EOC1. E repetindo o mesmo padrão de queima por EC do benchmark (Tabela 5.7), os maiores desvios estão na fronteira com o refletor, como destacados nas tabelas.

Os valores da concentração crítica de boro usando a teoria de 1 grupo modificado estão nas Tabelas 5.17 e 5.18. Na última coluna estão as diferenças com os valores para 2 grupos de energia, mostrando um aumento gradual dessa diferença ao longo do ciclo, chegando a 19 e 18ppm para o último instante de queima para os Núcleos 93 e 70, respectivamente. Baseado nesta Tabela, foi determinada a curva de boro dos Núcleos 93 e 70, Figuras 5.8 e 5.9.

⁸Erro relativo da queima média: $[(B_{EC(2D2G)} - B_{EC(2D1GM)})/B_{EC(2D2G)}] \times 100$.

Tabela 5.12: Distribuição de potência normalizada em BOC1 antes do equilíbrio do xenônio do Núcleo 70.

1,032	1,134	1,408	1,341	1,371	1,007	0,869	0,285
0,981	1,092	1,388	1,332	1,392	0,994	0,847	0,291
4,94%	3,70%	1,42%	0,67%	1,53%	1,29%	2,53%	-2,10%
1,134	1,228	1,740	1,460	1,749	1,149	0,640	0,259
1,092	1,214	1,632	1,500	1,704	1,184	0,652	0,262
3,70%	1,14%	6,21%	-2,74%	2,57%	-3,05%	-1,87%	-1,16%
1,408	1,740	1,705	2,064	2,043	1,349	0,627	0,221
1,388	1,632	1,754	2,017	2,105	1,329	0,638	0,217
1,42%	6,21%	-2,87%	2,28%	-3,03%	1,48%	-1,75%	1,81%
1,341	1,460	2,064	1,738	1,648	0,843	0,429	0,158
1,332	1,500	2,017	1,832	1,628	0,882	0,435	0,161
0,67%	-2,74%	2,28%	-5,40%	-1,21%	-4,62%	-1,40%	-1,9%
1,371	1,749	2,043	1,648	0,999	0,544	0,266	2D2G 2D1GM Desvio
1,392	1,704	2,105	1,628	1,038	0,534	0,269	
1,53%	2,57%	-3,03%	-1,21%	-3,90%	1,84%	-1,13%	
1,007	1,149	1,349	0,843	0,544	0,292	0,117	
0,994	1,184	1,329	0,882	0,534	0,296	0,120	
1,29%	-3,05%	1,48%	-4,62%	1,84%	-1,37%	-2,56%	
0,869	0,640	0,627	0,429	0,266	0,117		
0,847	0,652	0,638	0,435	0,269	0,120		
2,53%	-1,87%	-1,75%	-1,40%	-1,13%	-2,56%		
0,285	0,259	0,221	0,158				
0,291	0,262	0,217	0,161				
-2,10%	-1,16%	1,81%	-1,9%				

As concentrações médias estão nas Tabela 5.19 e 5.20, onde a última coluna mostra os desvios da teoria de 1 grupo modificado de energia em relação à de 2 grupos.

Por fim, nas Tabelas 5.21 e 5.22 estão os resultados do tempo de processamento para as duas teorias e para 1 grupo modificado otimizado, repetindo o mesmo padrão da Tabela 5.10.

Tabela 5.13: Distribuição de potência normalizada em EOC1 do Núcleo 93.

1,207	1,070	1,079	1,244	1,130	1,149	1,059	0,877
1,185	1,063	1,072	1,213	1,119	1,134	1,067	0,875
1,82%	0,65%	0,65%	2,49%	-0,97%	1,30%	-0,75%	0,23%
1,070	1,024	1,083	1,055	1,129	1,112	1,114	0,720
1,063	1,020	1,075	1,050	1,113	1,108	1,113	0,752
0,65%	-0,39%	0,74%	0,47%	1,41%	0,36%	0,09%	-4,44%
1,079	1,083	1,229	1,099	1,125	1,316	1,253	0,704
1,072	1,075	1,201	1,086	1,112	1,287	1,241	0,729
0,65%	0,74%	2,28%	1,18%	1,15%	2,20%	0,96%	-3,55%
1,244	1,055	1,099	1,048	1,064	1,062	0,963	0,500
1,213	1,050	1,086	1,036	1,055	1,062	0,974	0,539
2,49%	0,47%	1,18%	1,14%	0,84%	0,00%	-1,14%	-7,8%
1,130	1,129	1,125	1,064	1,251	0,989	0,698	2D2G 2D1GM Desvio
1,119	1,113	1,112	1,055	1,217	0,997	0,727	
-0,97%	1,41%	1,15%	0,84%	2,71%	-0,80%	-4,15%	
1,149	1,112	1,316	1,062	0,989	0,881	0,466	
1,134	1,108	1,287	1,062	0,997	0,896	0,503	
1,30%	0,36%	2,20%	0,00%	-0,80%	-1,70%	-7,94%	
1,059	1,114	1,253	0,963	0,698	0,466	2D2G 2D1GM Desvio	
1,067	1,113	1,241	0,974	0,727	0,503		
-0,75%	0,09%	0,96%	-1,14%	-4,15%	-7,94%		
0,877	0,720	0,704	0,500				
0,875	0,752	0,729	0,539				
0,23%	-4,44%	-3,55%	-7,8%				

Tabela 5.14: Distribuição de potência normalizada em EOC1 do Núcleo 70.

0,965	0,980	1,046	1,008	1,095	1,127	1,213	0,666
0,957	0,972	1,038	0,996	1,083	1,117	1,189	0,694
0,83%	0,82%	0,76%	1,19%	1,09%	0,88%	1,98%	-4,20%
0,980	0,997	1,194	1,032	1,255	1,134	1,006	0,652
0,972	0,995	1,163	1,027	1,223	1,126	1,014	0,679
0,82%	0,20%	2,60%	0,48%	2,55%	0,70%	-0,79%	-4,14%
1,046	1,194	1,081	1,239	1,285	1,294	1,037	0,641
1,038	1,163	1,076	1,208	1,263	1,265	1,044	0,663
0,76%	2,60%	0,46%	2,50%	1,71%	2,24%	-0,67%	-3,43%
1,008	1,032	1,239	1,131	1,299	1,075	0,899	0,504
0,996	1,027	1,208	1,123	1,268	1,080	0,920	0,540
1,19%	0,48%	2,50%	0,71%	2,39%	-0,46%	-2,33%	-7,14%
1,095	1,255	1,285	1,299	1,148	1,063	0,752	
1,083	1,223	1,263	1,268	1,148	1,068	0,783	
1,09%	2,55%	1,71%	2,39%	0,00%	-0,47%	-4,12%	
1,127	1,134	1,294	1,075	1,063	0,857	0,471	
1,117	1,126	1,265	1,080	1,068	0,887	0,513	
0,88%	0,70%	2,24%	-0,46%	-0,47%	-3,50%	-8,92%	
1,213	1,006	1,037	0,899	0,752	0,471		
1,189	1,014	1,044	0,920	0,783	0,513		
1,98%	-0,79%	-0,67%	-2,33%	-4,12%	-8,92%		
0,666	0,652	0,641	0,504				2D2G
0,694	0,679	0,663	0,540				2D1GM
-4,20%	-4,14%	-3,43%	-7,14%				Desvio

Tabela 5.15: Queima Média por EC em EOC1 do Núcleo 93.

21,049	17,411	17,866	20,631	17,357	15,624	12,843	9,415
20,510	17,104	17,764	19,862	17,314	15,372	13,101	9,341
-2,30%	1,76%	0,57%	3,72%	-0,25%	1,61%	-2,01%	0,78%
17,411	16,665	17,585	16,793	16,933	15,503	13,697	7,684
17,104	16,608	17,288	16,812	16,630	15,667	13,717	8,103
1,76%	0,34%	1,69%	-0,11%	1,79%	-1,06%	-0,15%	-5,45%
17,866	17,585	21,102	17,255	17,281	19,199	16,520	7,449
17,764	17,288	20,574	16,977	17,230	18,757	16,556	7,699
0,57%	1,69%	2,50%	1,61%	-0,29%	2,30%	-0,22%	-3,35%
20,631	16,793	17,255	16,211	15,905	14,483	11,289	5,048
19,862	16,812	16,977	16,137	15,929	14,666	11,426	5,445
3,72%	-0,11%	1,61%	0,46%	-0,15%	-1,26%	-1,21%	3,64%
17,357	16,933	17,281	15,905	18,696	12,582	7,557	
17,314	16,630	17,230	15,929	18,232	12,809	7,890	
-0,25%	1,79%	-0,29%	-0,15%	2,48%	-1,80%	-4,41%	
15,624	15,503	19,199	14,483	12,582	10,122	4,611	
15,372	15,667	18,757	14,666	12,809	10,316	4,949	
1,61%	-1,06%	2,30%	-1,26%	-1,80%	-1,92%	-7,33%	
12,843	13,697	16,520	11,289	7,557	4,611		
13,101	13,717	16,556	11,426	7,890	4,949		
-2,01%	-0,15%	-0,22%	-1,21%	-4,41%	-7,33%		
9,415	7,684	7,449	5,048				2D2G
9,341	8,103	7,699	5,445				2D1GM
0,78%	-5,45%	-3,35%	3,64%				Desvio

Tabela 5.16: Queima Média por EC em EOC1 do Núcleo 70.

14,135	14,812	16,850	16,245	17,377	15,859	15,633	6,934
13,895	14,611	16,718	16,104	17,377	15,777	15,395	7,249
-1,70%	1,02%	0,78%	0,87%	0,00%	0,52%	1,52%	-4,54%
14,812	15,488	19,958	17,051	20,785	16,552	12,257	6,555
14,611	15,445	19,199	17,160	20,263	16,693	12,481	6,819
1,02%	0,28%	3,80%	-0,64%	2,51%	-0,85%	-1,83%	-4,03%
16,850	19,958	18,590	21,987	22,133	18,880	12,197	5,998
16,718	19,199	18,701	21,344	22,092	18,530	12,387	6,137
0,78%	3,80%	-0,60%	2,92%	0,18%	1,85%	-1,56%	-2,32%
16,245	17,051	21,897	19,171	20,354	13,840	9,566	4,528
16,104	17,160	21,344	19,411	19,944	14,132	9,823	4,813
0,87%	-0,64%	2,92%	-1,25%	-0,15%	2,01%	-2,69%	-6,29%
17,377	20,785	22,133	20,354	15,325	11,536	7,105	
17,377	20,263	22,092	19,944	15,542	11,543	7,387	
0,00%	2,51%	0,18%	-0,15%	-1,41%	-0,06%	-3,97%	
15,859	16,552	18,880	13,840	11,536	7,988	3,882	
15,777	16,693	18,530	14,132	11,543	8,228	4,184	
0,52%	-0,85%	1,85%	2,01%	-0,06%	-3,00%	-7,78%	
15,663	12,257	12,197	9,566	7,185	3,882		
15,395	12,481	12,387	9,823	7,387	4,184		
1,52%	-1,83%	-1,56%	-2,69%	-3,97%	-7,78%		
6,934	6,555	5,998	4,528				2D2G
7,249	6,819	6,137	4,813				2D1GM
-4,54%	-4,03%	-2,32%	-6,29%				Desvio

Tabela 5.17: Concentração crítica de boro $\times$ queima média do Núcleo 93.

Instante DEPP (dias)	Queima Média (MWD/KgU)	CB-Calculado 2D2G (ppm)	CB-Calculado 2D1GM (ppm)	Diferença (ppm)
0	0	1511,6 (sem Xe)	1518,2	6,6
0	0	1258,6 (Xe equilíbrio)	1265,9	7,3
30	1,098	1287,9	1291,7	3,8
60	2,197	1240,6	1245,6	5,0
90	3,295	1155,3	1161,4	6,1
120	4,394	1050,9	1058,5	7,6
150	5,492	936,0	945,3	9,3
180	6,590	815,7	826,4	10,7
210	7,689	692,9	705,1	12,2
240	8,787	568,7	582,3	13,6
270	9,886	445,2	459,9	14,7
300	10,984	323,1	339,0	15,9
330	12,083	202,6	219,7	17,1
360	13,181	84,7	102,6	17,9
381,49	13,968	1,7	20,4	18,7

Tabela 5.18: Concentração crítica de boro $\times$ queima média do Núcleo 70.

Instante DEPP (dias)	Queima Média (MWD/KgU)	CB-Calculado 2D2G (ppm)	CB-Calculado 2D1GM (ppm)	Diferença (ppm)
0	0	1832,3 (sem Xe)	1836,9	4,6
0	0	1553,4 (Xe equilíbrio)	1558,5	5,1
30	1,098	1565,5	1567,5	2,0
60	2,197	1500,5	1501,7	1,2
90	3,295	1394,7	1395,9	1,2
120	4,394	1268,9	1270,9	2,0
150	5,492	1134,6	1138,3	3,7
180	6,590	998,3	1003,6	5,3
210	7,689	861,9	869,1	7,2
240	8,787	727,3	736,6	9,3
270	9,886	595,5	606,5	11,0
300	10,984	466,1	478,8	12,7
330	12,083	339,6	353,9	14,3
360	13,181	216,4	232,1	15,7
381,49	13,968	128,8	146,3	17,5

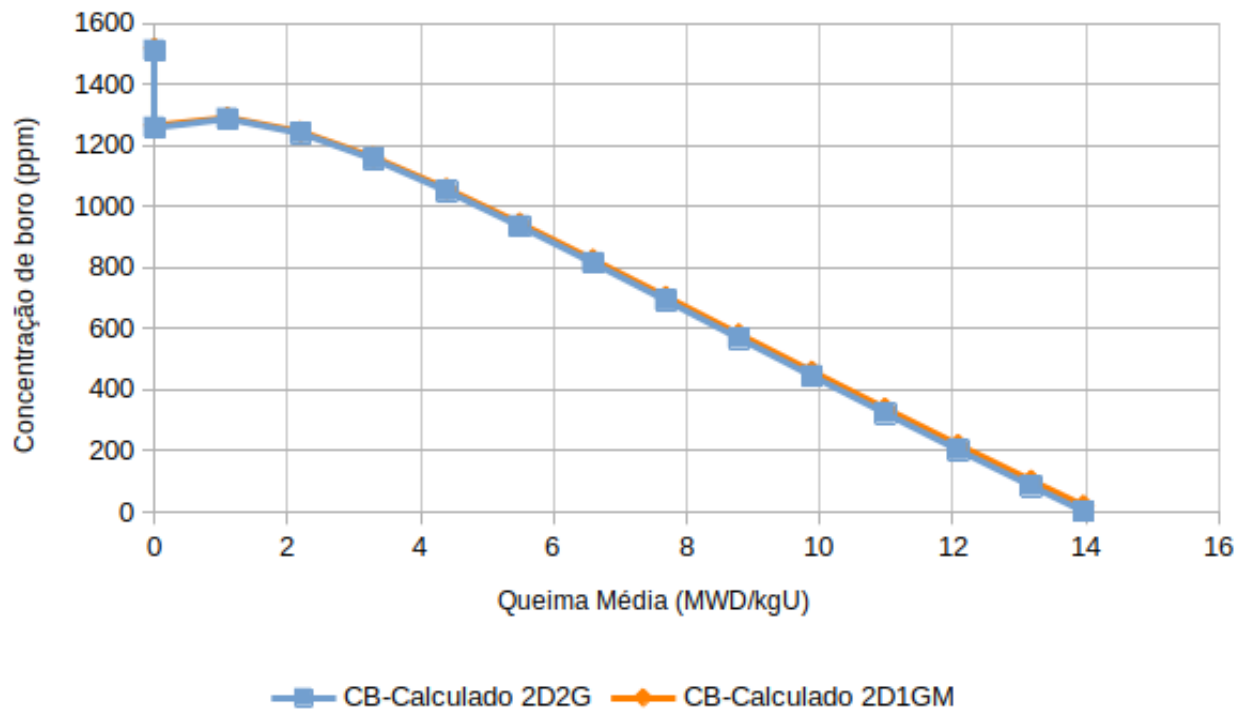


Figura 5.8: Curva de Boro do Núcleo 93.

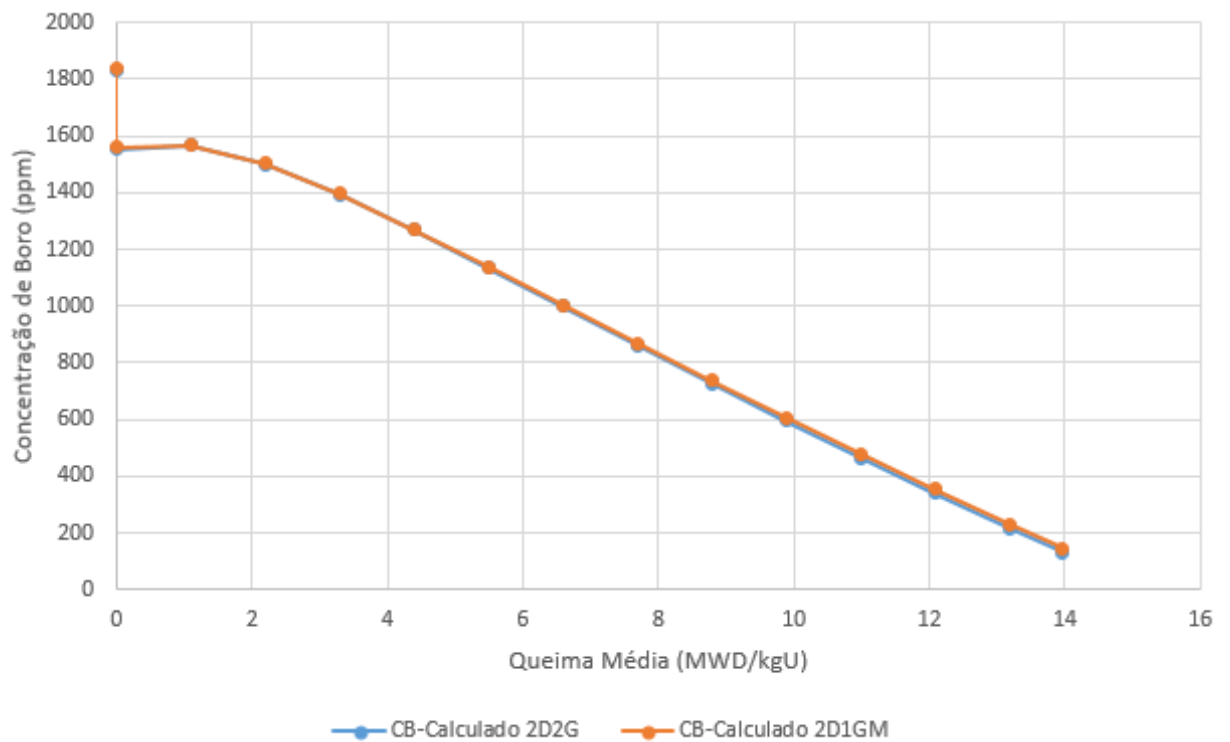


Figura 5.9: Curva de Boro do Núcleo 70.

Tabela 5.19: Concentrações Médias em EOC1 do Núcleo 93.

Nuclídeo	2D2G ($10^{24} \text{atomos/cm}^3$)	2D1GM ($10^{24} \text{atomos/cm}^3$)	Desvio (%)
U^{235}	$0,84489 \times 10^{-4}$	$0,84375 \times 10^{-4}$	0,13
U^{236}	$0,13515 \times 10^{-4}$	$0,13538 \times 10^{-4}$	-0,17
U^{238}	$0,62198 \times 10^{-2}$	$0,62198 \times 10^{-2}$	0,00
Pu^{239}	$0,24516 \times 10^{-4}$	$0,24592 \times 10^{-4}$	-0,31
Pu^{240}	$0,64472 \times 10^{-5}$	$0,64677 \times 10^{-5}$	-0,32
Pu^{241}	$0,39730 \times 10^{-5}$	$0,39624 \times 10^{-5}$	0,27
Pu^{242}	$0,72523 \times 10^{-6}$	$0,71633 \times 10^{-6}$	1,23
Xe^{135}	$0,20057 \times 10^{-8}$	$0,20108 \times 10^{-8}$	-0,25
PF	$0,95750 \times 10^{-4}$	$0,95751 \times 10^{-4}$	-0,001
VQ	$0,49788 \times 10^{-6}$	$0,50336 \times 10^{-6}$	-1,10

Tabela 5.20: Concentrações Médias em EOC1 do Núcleo 70.

Nuclídeo	2D2G ($10^{24} \text{atomos/cm}^3$)	2D1GM ($10^{24} \text{atomos/cm}^3$)	Desvio (%)
U^{235}	$0,90716 \times 10^{-4}$	$0,90624 \times 10^{-4}$	0,10
U^{236}	$0,13829 \times 10^{-4}$	$0,13849 \times 10^{-4}$	-0,14
U^{238}	$0,62135 \times 10^{-2}$	$0,62135 \times 10^{-2}$	0,00
Pu^{239}	$0,24531 \times 10^{-4}$	$0,24615 \times 10^{-4}$	-0,34
Pu^{240}	$0,62416 \times 10^{-5}$	$0,62616 \times 10^{-5}$	-0,32
Pu^{241}	$0,38885 \times 10^{-5}$	$0,38769 \times 10^{-5}$	0,3
Pu^{242}	$0,69409 \times 10^{-6}$	$0,68694 \times 10^{-6}$	0,64
Xe^{135}	$0,20589 \times 10^{-8}$	$0,20648 \times 10^{-8}$	-0,29
PF	$0,95741 \times 10^{-4}$	$0,95743 \times 10^{-4}$	-0,002
VQ	$0,41843 \times 10^{-6}$	$0,42929 \times 10^{-6}$	-2,59

Tabela 5.21: Resultados para o tempo de processamento do Núcleo 93.

Teoria	Tempo (min)
2 grupos	59
1 grupo modificado	28
1 grupo modificado (otimizado)	7

Tabela 5.22: Resultados para o tempo de processamento do Núcleo 70.

Teoria	Tempo (min)
2 grupos	58
1 grupo modificado	28
1 grupo modificado (otimizado)	8

II) Resultados para todas as configurações

Os resultados para a concentração de boro das 100 configurações geradas, variando os instantes de queima com intervalos de 120 DEPP (compreendendo todo o ciclo, de 0 a 381,49 DEPP) estão nas Figuras de 5.10 a 5.14.

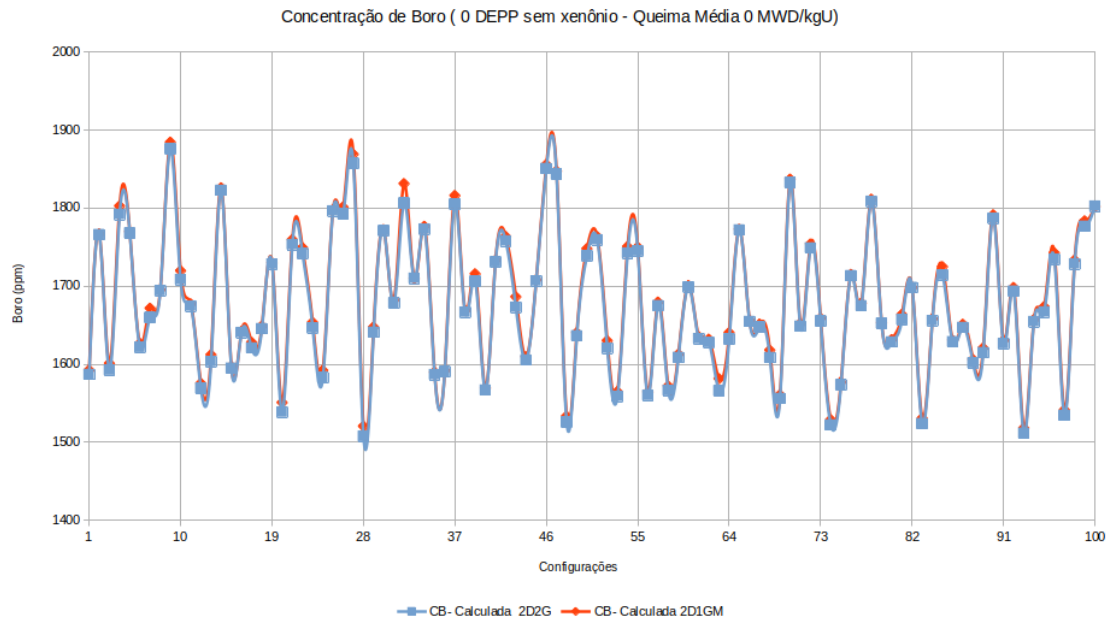


Figura 5.10: Concentração de Boro - 0 DEPP.

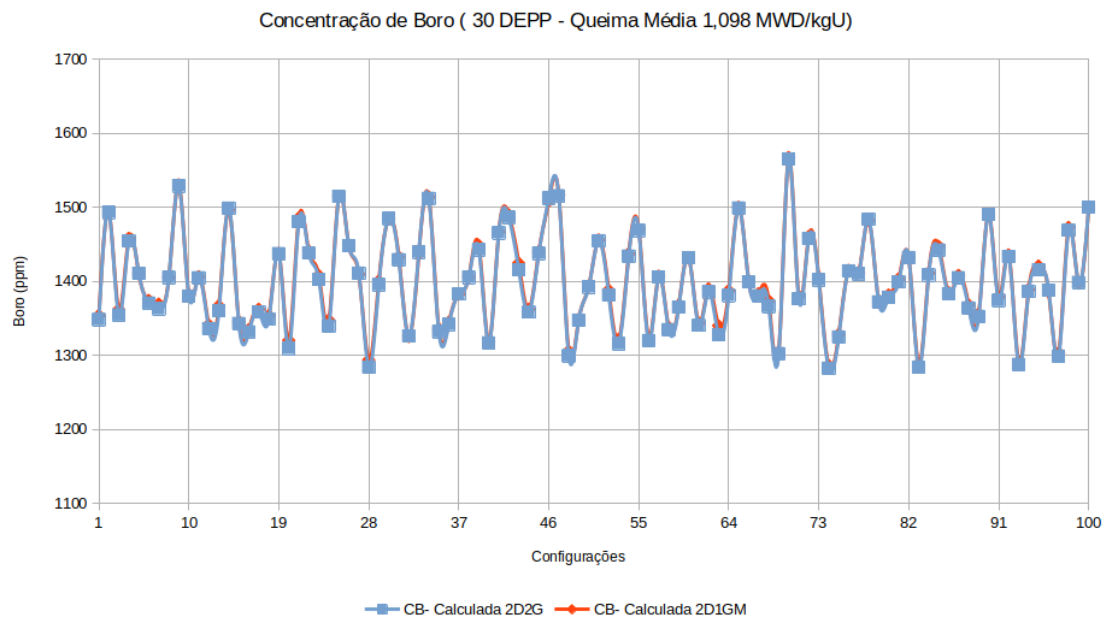


Figura 5.11: Concentração de Boro - 30 DEPP.

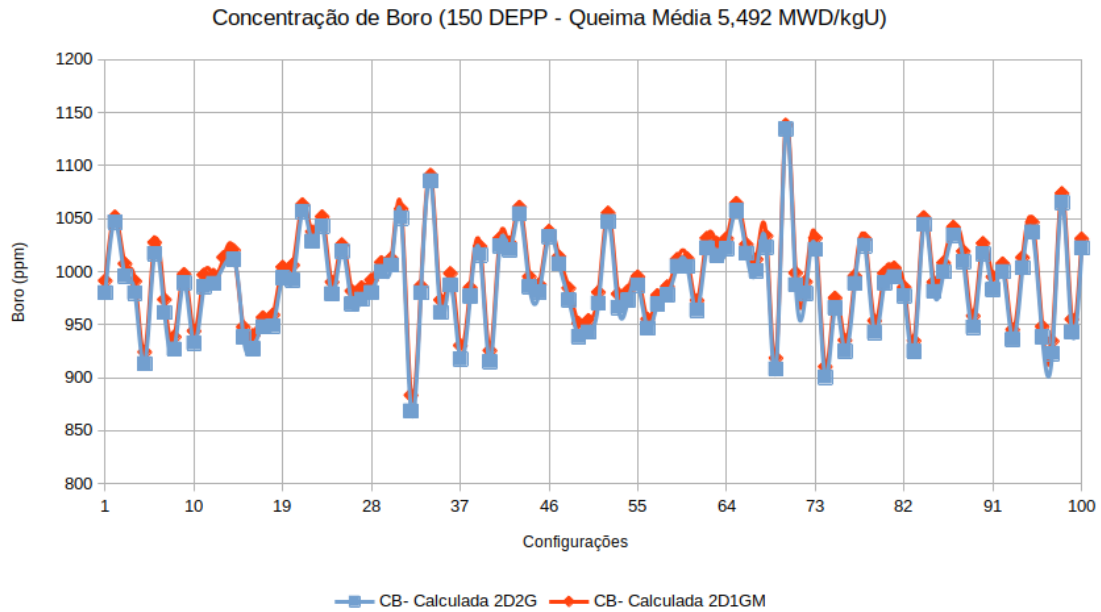


Figura 5.12: Concentração de Boro - 150 DEPP.

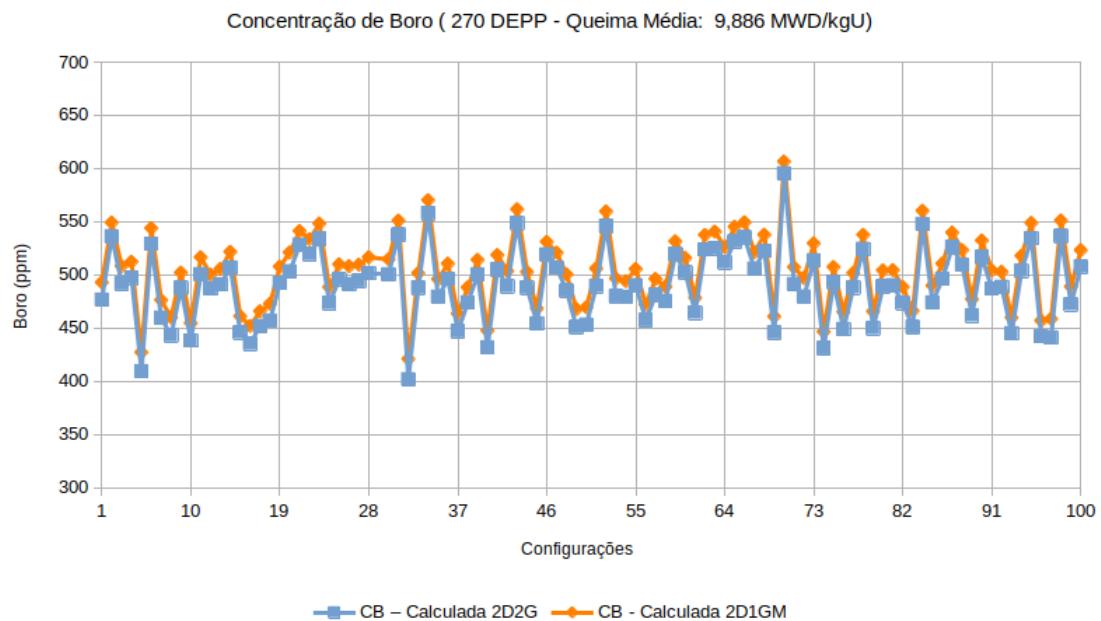


Figura 5.13: Concentração de Boro - 270 DEPP.

A Figura 5.14 mostra que os desvios da teoria de 1 grupo modificado para o último instante, 381,49 DEPP⁹, são maiores que os instantes anteriores, como mostrado nas Figuras 5.10 a 5.13. Este comportamento é esperado conforme observado

⁹Os núcleos, cuja concentração de boro foi negativa para o último instante, são considerados não aptos para recarga, uma vez que o comprimento do ciclo é menor que 381,49 DEPP, isto é, a concentração inicial de boro não é capaz de controlar a reatividade do núcleo por todo o ciclo considerado.

nas Tabelas 5.17 e 5.18, que mostra uma aumento gradual da diferença da concentração de boro calculada com as duas teorias ao longo do ciclo, com a maior diferença no último instante.

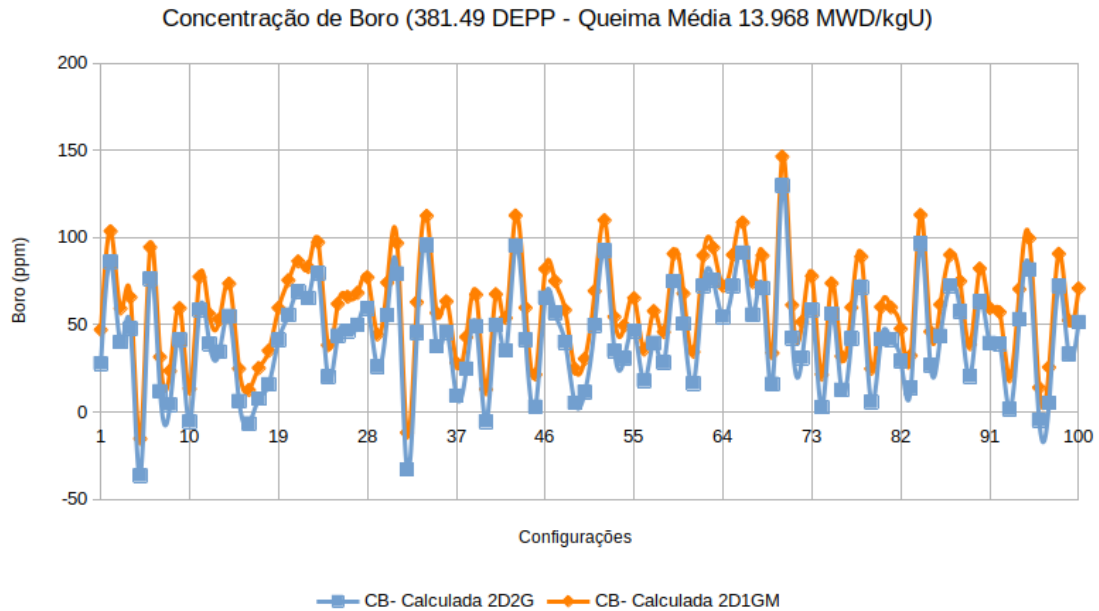


Figura 5.14: Concentração de Boro - 381,49 DEPP.

O tempo médio computacional para cada configuração, para as duas teorias, e para 1 grupo modificado otimizado são mostrados no gráfico da Figura 5.15, corroborando a redução do tempo computacional decorrente do que o uso da teoria de 1 grupo modificado, com e sem otimização, considerando 1 ciclo completo. Os tempos dos núcleos 93 e 70 ficaram abaixo da média (Tabelas 5.21 e 5.22), assim como o benchmark (Tabela 5.10).

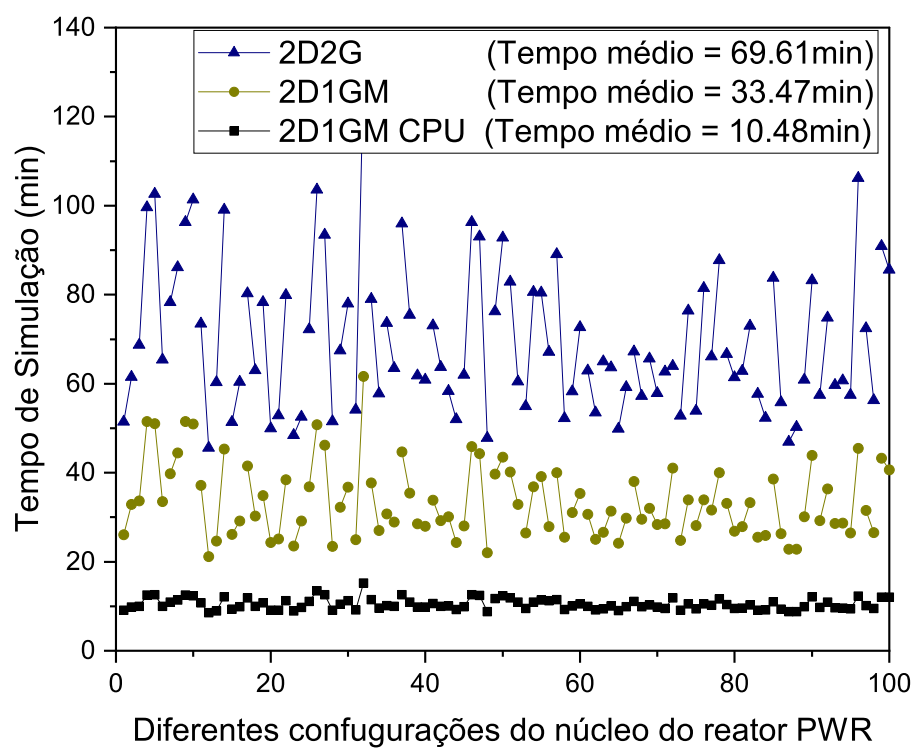


Figura 5.15: Tempo médio de processamento para 100 configurações de 1 ciclo completo.

Capítulo 6

Discussões e Conclusões

O trabalho desenvolvido nesta tese foi a aplicação da teoria da difusão de um grupo modificado de energia à equação da difusão de nêutrons, que, juntamente com otimizações no compilador, reduzem o tempo computacional dos cálculos neutrônicos na otimização de núcleo. Sabendo-se que neste processo vários tipos de carregamento deverão ser avaliados, o tempo computacional médio de cada avaliação deve ser reduzido de maneira a contornar o problema do tempo computacional empregado nos cálculos neutrônicos, que é o ponto nevrálgico no processo de otimização de núcleo: uma resposta num tempo razoável para um novo padrão de carregamento que posso substituir o núcleo ao fim do ciclo de queima.

Na primeira análise, os dois benchmarks, Biblis e Koebke, foram usados para avaliar o uso da teoria de 1 grupo modificado de energia quanto à precisão e o tempo computacional dos cálculos. Nessa etapa, os resultados do k_{eff} e da distribuição de potência normalizada, além dos tempos de processamento para cada metodologia foram comparados. As Tabelas 5.1 e 5.3 mostram que os resultados do k_{eff} usando 1 grupo modificado apresentam desvios menores que 1%. O tempo computacional empregado nas duas metodologias é reduzido para menos da metade (em comparação à referência, no caso, a de 2 grupos). E as otimizações reduziram ainda mais o tempo dos cálculos com 1 grupo modificado sem alteração dos resultados e do código desenvolvido.

As Tabelas 5.2 e 5.4 mostram a distribuição de potência normalizada dos benchmarks Biblis e Koebke, respectivamente. No caso do Biblis, o maior erro relativo porcentual foi de 4,8% na fronteira com o refletor, um comportamento que se repetirá na análise de 1 ciclo completo de queima. No caso do Koebke, o maior erro relativo porcentual foi de 5,04%, só que desta vez na região mais interna. O Koebke, que é um reator subcrítico, não apresenta erros maiores que 2% na fronteira com o refletor, pois ele próprio já possui uma quantidade de material físsil menor, não havendo uma diminuição tão abrupta de fluxo. Seus erros maiores que 5% estão nas regiões mais internas do núcleo devido a presença das barras de controle (elemen-

tos fortemente absorvedores), apresentando uma diminuição do fluxo de nêutrons devido à absorção, e, consequentemente, um erro na potência calculada para estas regiões.

Os resultados do k_{eff} e tempo de processamento apresentam o mesmo comportamento dos resultados do Biblis, permitindo concluir que, apesar de restrita a reatores de grande porte, a teoria de 1 grupo modificado pode ser aplicada a análise de reatores de pequeno porte, uma vez que os erros maiores são devidos à presença de barras de controle, não de seu tamanho.

Com relação à redução de tempo computacional do código otimizado, as Figuras 5.3 e 5.6 mostram que os resultados do tempo médio para 100 configurações estão coerentes com os resultados das Tabelas 5.1 e 5.3. As duas figuras mostram que as medidas para redução do tempo computacional foram bem sucedidas, mostrando ser capaz de acelerar os cálculos sem alteração dos resultados e das linhas de código, dispensando uma programação específica para o melhor desempenho do código.

Na análise de 1 ciclo completo, um benchmark cujas concentrações iniciais são conhecidas foi usado para avaliar o comportamento da teoria de 1 grupo modificado de energia nas etapas de pesquisa de criticalidade, queima e depleção. O benchmark foi analisado durante seu 1º ciclo de carregamento. A teoria de 1 grupo modificado não apresenta bons resultados para a distribuição de potência antes do equilíbrio de xenônio (BOC1), como podemos ver na Tabela 5.5 erros maiores que 9%. Mas na Tabela 5.6, observam-se resultados bem diferentes para a distribuição de potência ao final do ciclo (EOC1), com o maior desvio na fronteira com o refletor, 3,86%, repetindo o mesmo padrão de resultados que obtemos com o Biblis (Tabela 5.2). A queima média por EC também apresenta o mesmo comportamento, onde o maior desvio foi de 5,17% na fronteira com o refletor. Isso mostra que independente do reator a ser simulado o comportamento da teoria de 1 grupo modificado é sempre o mesmo.

O parâmetro de operação do reator, a concentração crítica de boro (C_B), foi calculada para um ciclo de queima com 381,49 DEPP, ou aproximadamente, 1 ano e 16,5 dias, dividido em intervalos de 30 DEPP (Tabela 5.8). Para cada instante de tempo e queima acumulada correspondente foi calculada a concentração de boro, em ppm, e construída a curva de boro para o benchmark, Figura 5.7. Os resultados da concentração de boro para 1 grupo modificado mostra um aumento gradual da diferença com a teoria de 2 grupos à medida que o ciclo avança, de tal forma que a diferença atinge, aproximadamente, 20ppm no último instante.

Os resultados para as concentrações dos núclídeos em EOC1 estão na Tabela 5.9, mostrando que o maior erro com 1 grupo modificado foi 1,28%.

Os tempos computacionais das duas teorias e com otimização estão na Tabela 5.10 e repetem o mesmo padrão das Tabelas 5.1 e 5.3 mostrando a eficácia das

metodologias utilizadas para redução do tempo computacional considerando 1 ciclo de queima completo.

Baseado na Figura 4.3, 100 configurações diferentes foram geradas embaralhando seus elementos combustíveis para análise do 1 ciclo de queima de cada uma. Os resultados apresentados correspondem às configurações com menor e maior concentração de boro em EOC1, 1,75 e 129ppm, respectivamente. A distribuição de potência normalizada em BOC1 antes do equilíbrio de xenônio e em EOC1 apresentam os mesmos comportamentos que o benchmark, com os erros relativos maiores localizados na fronteira com o refletor em EOC1 (Tabelas 5.11 a 5.14), assim como distribuição de queima, como mostram as Tabelas 5.15 e 5.16. Os erros apresentam o mesmo comportamento para qualquer configuração do benchmark.

A concentração crítica de boro também apresentou o mesmo comportamento do benchmark (Tabela 5.8). As Tabelas 5.17 e 5.18 mostram o gradual aumento da diferença entre a concentração de boro calculada usando a teoria de 1 grupo modificado e a de 2 grupos, atingindo valores de até 19ppm. As curvas de boro das duas configurações também foram construídas, como mostram as Figuras 5.8 e 5.9.

Os tempos computacionais das duas configurações estão nas Tabelas 5.21 e 5.22, mostrando que também seguiram a mesma tendência dos resultados anteriores relacionados ao desempenho computacional da teoria de 1 grupo modificado.

Como forma de abranger as conclusões do comportamento da teoria de 1 grupo modificado para as 100 configurações, as Figuras 5.10, 5.11, 5.12, 5.13 e 5.14, mostram os valores calculados da concentração de boro para diferentes instantes de tempo, para as duas teorias, com as queimas acumuladas correspondentes, com a última Figura (5.14) corroborando o maior desvio da concentração de boro calculada para 1 grupo modificado para os instantes finais do ciclo.

A Figura 5.15 reúne todos os tempos computacionais das 100 configurações usando as duas teorias e 1 grupo modificado otimizado. Como esperado, o tempo médio para cada configuração seguiu a tendência dos resultados anteriores, mostrando que o uso da teoria de 1 grupo modificado otimizado contornou o principal problema computacional do processo de otimização do núcleo: redução do tempo médio de avaliação de cada configuração, de forma que o tempo total para um dado número de configurações ainda seja razoável dentro do cronograma de recarga da usina nuclear.

Sugestões para trabalhos futuros

- Com relação à aplicabilidade da teoria de 1 grupo modificado em reatores de pequeno porte, simular um reator que seja crítico (o Koebeke é subcrítico), para 1 ciclo completo de queima;

- Ao invés de otimizações no compilador, investigar a redução dos tempos computacionais com o emprego de paralelização direta no código (explícita) e comparar com os resultados de tempo desta tese;
- Em relação ao Modelo de Programação Paralela, ao invés do OpenMP, usar o modelo MPI, usando o mesmo tipo de computador e comparar com os resultados desta tese. O modelo de Programação paralela é uma biblioteca, não uma linguagem de computação. Para isso será necessário escrever o código no FORTRAN 90 e compilá-lo para vincular à biblioteca de comunicações necessárias entre os processadores;
- Considerando a arquitetura da máquina, usar a paralelização na GPU (*Graphic Processing Unit*) para aceleração dos tempos computacionais. A GPU é uma evolução da CPU, proporcionando maiores ganhos em termos de tempo computacional, pois possuem capacidades de processamento muito maiores que a CPU. Com a popularização de processadores da marca NVIDIA, este tipo de arquitetura com múltiplos núcleos fez com que a GPU assumisse um protagonismo em relação ao poder de processamento, abrindo a possibilidade de aceleração dos tempos computacionais ainda melhores;
- Para diminuir o esforço computacional, realizar os cálculos com malha grossa, ao invés de malha fina, simulando até o equilíbrio do xenônio, ao invés de simular 1 ciclo completo de queima.

Referências Bibliográficas

- GOLDBARG, M. C., L. H. P. L. *Otimização Combinatória e Programação Linear: Modelos e Algoritmos*. Editora Campos, Rio de Janeiro, 2005.
- LI, Z., WANG, J., DING, M. “A review on optimization methods for nuclear reactor fuel reloading analysis”, *Nuclear Engineering and Design*, v. 397, pp. 111950, 2022.
- DA SILVA, A. C., MARTINEZ, A. S., LIMA, M. A. C., et al. “Simplified model for nuclear reactor core loading pattern optimization”, *Nuclear Engineering and Design*, v. 414, pp. 112622, 2023.
- OLIVEIRA, I. M. S. *Inteligência de Enxames Aplicado ao Problema de Otimização de Recargas de Reatores Nucleares a Água Pressurizada*. Tese de doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2013.
- WAINTRAUB, M. *Algoritmos Paralelos de Otimização por Enxames de Partículas em Problemas Nucleares*. Tese de doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2009.
- DUDERSTADT, JAMES J., H. L. J. *Nuclear Reactor Analysis*. Wiley, 1976.
- NAKAMURA, S. *Computational Methods in Engineering and Science*. John Wiley & Sons, New York, 1977.
- BURDEN, R. L., FAIRES, J. D., BURDEN, A. M. *Numerical Analysis*. Cengage Learning, Boston, Massachusetts, 2015.
- BARNEY, B., OTHERS. “Introduction to parallel computing”, *Lawrence Livermore National Laboratory*, v. 6, n. 13, pp. 10, 2010.
- PAHADE, P., DAWALE, M. “Introduction to Compiler and its Phases”, *International Research Journal of Engineering and Technology (IRJET)*, v. 6, n. 1, pp. 1318–1322, 2019.
- MOGENSEN, T. Æ. *Basics of compiler design*. Torben Ægidius Mogensen, 2009.

- INTEL. “Quick Reference Guide to Optimization with Intel® C++ and Fortran Compilers v19.1.” <https://www.intel.com/content/www/us/en/content-details/671224/quick-reference-guide-to-optimization-with-intel-c-and-fortran-compilers.html>, 2018. Acessado em 10 Março de 2024.
- EIGENMANN, R., HOEFLINGER, J. “Parallelizing and vectorizing compilers”, *Proceedings of the IEEE*, 2000.
- BIK, A. J., KREITZER, D. L., TIAN, X. “A case study on compiler optimizations for the intel® Core™ 2 duo processor”, *International Journal of Parallel Programming*, v. 36, n. 6, pp. 571–591, 2008.
- BERGMANN, S. D. “Compiler design: theory, tools, and examples”, 2017.
- INTEL. “Intel® Fortran Compiler Developer Guide and Reference”. <https://www.intel.com/content/www/us/en/docs/fortran-compiler/developer-guide-reference/2025-0/overview.html>, 2024. Acessado em 19 Outubro de 2024.
- MELHEM, R. “Introduction to parallel computing”, 1992.
- CHANDRA, R. *Parallel programming in OpenMP*. Morgan kaufmann, 2001.
- MÜLLER, E. Z., WEISS, Z. J. “Benchmarking with the multigroup diffusion high-order response matrix method”, *Annals of Nuclear Energy*, v. 18, pp. 535–544, 1991. doi: [https://doi.org/10.1016/0306-4549\(91\)90098-I](https://doi.org/10.1016/0306-4549(91)90098-I).
- TRKOV, A., RAVNIK, M. “Effective Diffusion Homogenization of Cross Section for Pressurized Water Reactor Core Calculations”, *Nuclear Science and Engineering*, v. 116, pp. 86–95, 1994.
- CAPOSSOLI, E. F. *Análise Sistemática da equação da difusão de nêutrons utilizando a teoria de um grupo modificado de energia*. Tese de Mestrado, COPPE/UFRJ, 2009.
- KOEBKE, K., WAGNER, M., WINTER, H., et al. “PWR Burnup Benchmark Problem”, *ANL-7416, Supplement*, v. 3, pp. 880–924, 1984.
- BEAN, J. C. “Genetic Algorithms and Random Keys for Sequencing and Optimization”, *ORSA JOURNAL OF COMPUTING*, v. 6, n. 2, pp. 154–160, 1994.
- DA SILVA, A. C. *Simulação da Multiplicação Subcrítica de Nêutrons Através da Solução da Equação Híbrida da Difusão com Fontes Externas de Nêutrons*. Tese de Doutorado, Universidade Federal do Rio de Janeiro, 2011.

ALVIM, A. C. M., DA SILVA, F. C., MARTINEZ, A. S. “Depletion calculation for a nodal reactor physics code”. In: *International Conference on Nuclear Engineering*, v. 49309, pp. 311–316, 2010.

Apêndice A

Equações de Depleção

Neste apêndice são demonstradas as equações de depleção dos nuclídeos (ALVIM *et al.*, 2010) da cadeia de decaimento da Figura A.1. A Equação A.1 mostra a fórmula geral de depleção para nuclídeos numerados de 1 a 8. E, de acordo com a fórmula, as equações de depleção isotópicas dos nuclídeos numerados de 1 a 8 são:

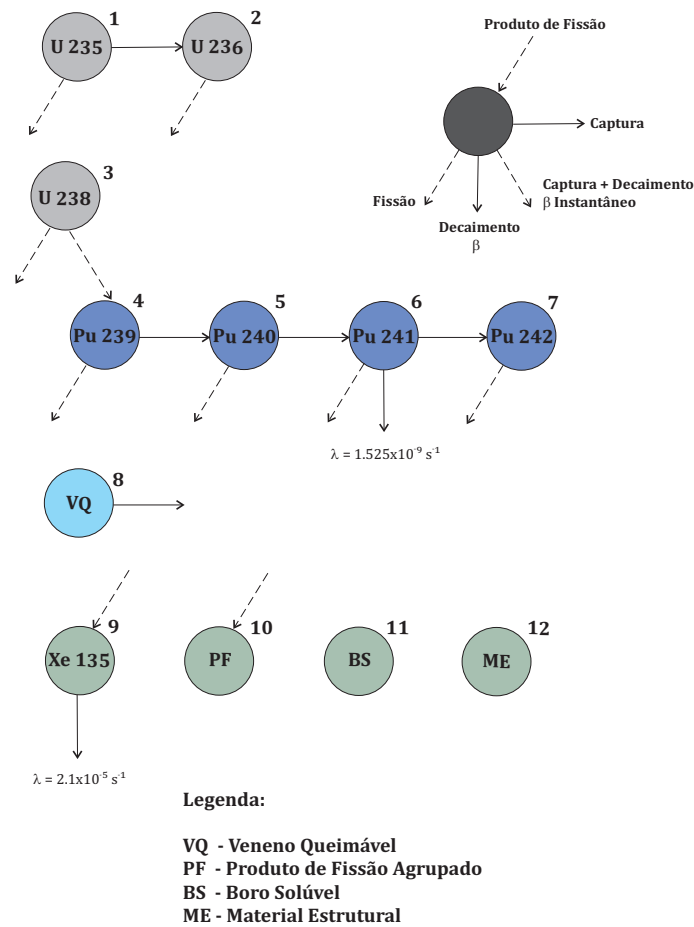


Figura A.1: Cadeia usada para equações de depleção.

$$\frac{dN_j(t)}{dt} + \left\{ \sum_g \sigma_{aj}^g \phi_g + \lambda_j \right\} N_j(t) = N_{j-1}(t) \left\{ \sum_g \sigma_{cj-1}^g \phi_g \right\} \quad (\text{A.1})$$

- $j = 1 \rightarrow U^{235}$:

$$\frac{dN_1(t)}{dt} = - \left\{ \sum_{g=1}^G \sigma_{a1}^g \phi_g \right\} N_1(t) \quad (\text{A.2})$$

- $j = 2 \rightarrow U^{236}$:

$$\frac{dN_2(t)}{dt} = - \left\{ \sum_{g=1}^G \sigma_{a2}^g \phi_g \right\} N_2(t) + N_1(t) \sum_{g=1}^G \sigma_{c1}^g \phi_g \quad (\text{A.3})$$

- $j = 3 \rightarrow U^{238}$:

$$\frac{dN_3(t)}{dt} = - \left\{ \sum_{g=1}^G \sigma_{a3}^g \phi_g \right\} N_3(t) \quad (\text{A.4})$$

- $j = 4 \rightarrow Pu^{239}$:

$$\frac{dN_4(t)}{dt} = - \left\{ \sum_{g=1}^G \sigma_{a4}^g \phi_g \right\} N_4(t) + N_3(t) \sum_{g=1}^G \sigma_{c3}^g \phi_g \quad (\text{A.5})$$

- $j = 5 \rightarrow Pu^{240}$:

$$\frac{dN_5(t)}{dt} = - \left\{ \sum_{g=1}^G \sigma_{a5}^g \phi_g \right\} N_5(t) + N_4(t) \sum_{g=1}^G \sigma_{c4}^g \phi_g \quad (\text{A.6})$$

- $j = 6 \rightarrow Pu^{241}$:

$$\frac{dN_6(t)}{dt} = - \left\{ \sum_{g=1}^G \sigma_{a6}^g \phi_g \right\} N_6(t) - \lambda_6 N_6(t) + N_5(t) \sum_{g=1}^G \sigma_{c5}^g \phi_g \quad (\text{A.7})$$

- $j = 7 \rightarrow Pu^{242}$:

$$\frac{dN_7(t)}{dt} = - \left\{ \sum_{g=1}^G \sigma_{a7}^g \phi_g \right\} N_7(t) + N_6(t) \sum_{g=1}^G \sigma_{c6}^g \phi_g \quad (\text{A.8})$$

- $j = 8 \rightarrow \text{Veneno Queimável (VQ)}$:

$$\frac{dN_8(t)}{dt} = - \left\{ \sum_{g=1}^G \sigma_{c8}^g \phi_g \right\} N_8(t) \quad (\text{A.9})$$

Para o nuclídeo $j = 9$ (Xe^{135}) a concentração de equilíbrio é determinada pela Equação A.10:

- $j = 9 \rightarrow Xe^{135}$:

$$N_{Xe}^{\infty} = \frac{\sum_{j=1}^7 \gamma_{Xe}^j N_j(t) \sum_{g=1}^G \sigma_{fj}^g \phi_g}{\sum_{g=1}^G \sigma_{cXe}^g + \lambda_{Xe}} \quad (A.10)$$

A equação de depleção dos produtos de fissão (PF), $j = 10$, é dada pela equação A.11:

- $j = 10 \rightarrow$ Produtos de Fissão (PF):

$$\frac{dN_{PF}(t)}{dt} = \sum_{j=1}^7 \gamma_{PF}^j N_j(t) \sum_{g=1}^G \sigma_{fj}^g \phi_g \quad (A.11)$$

O boro solúvel ($j = 11$) e o material estrutural ($j = 12$) não são depletáveis.

Apêndice B

Pesquisa de Criticalidade

A pesquisa de criticalidade do núcleo do reator será feito utilizando como parâmetro a concentração de boro solúvel. Baseado na hipótese que este parâmetro é uma função linear do k_{eff} , o cálculo da concentração de boro, em partes por milhão (ppm), é feito dentro da rotina *AjustaConcentracaoBoro* usando dois cálculos de fluxos, para dois valores de concentração de boro de pesquisa, tal que:

$$[CB] = \frac{[CB]^{(2)} - [CB]^{(1)}}{k_{eff}^{(2)} - k_{eff}^{(1)}} k_{eff} + \frac{[CB]^{(1)} k_{eff}^{(2)} - [CB]^{(2)} k_{eff}^{(1)}}{k_{eff}^{(2)} - k_{eff}^{(1)}}, \quad (B.1)$$

onde os sobrescritos 1 e 2 relacionam os valores de $k_{eff}^{(1)}$ e $k_{eff}^{(2)}$ com os valores calculados de $[CB]^{(1)}$ e $[CB]^{(2)}$, respectivamente.

A escolha do valor do parâmetro $[CB]$ é feita fazendo-se $k_{eff} = 1$ (condição de criticalidade) para o próximo cálculo de fluxo na Equação (B.1). E, desta forma, quando consideramos a i -ésima iteração de pesquisa, chega-se à expressão geral do parâmetro [ppm]:

$$[CB]^{i+1} = \frac{(1 - k_{eff}^{(i-1)})[CB]^i - (1 - k_{eff}^{(i)})[CB]^{(i-1)}}{k_{eff}^{(i)} - k_{eff}^{(i-1)}}, \quad i = 1, 2, 3, \dots, N, \quad (B.2)$$

onde $k_{eff}^{(i)}$ e $[CB]^i$ representam o fator de multiplicação e a concentração de boro da i -ésima iteração, respectivamente.

Apêndice C

Opções de Otimização

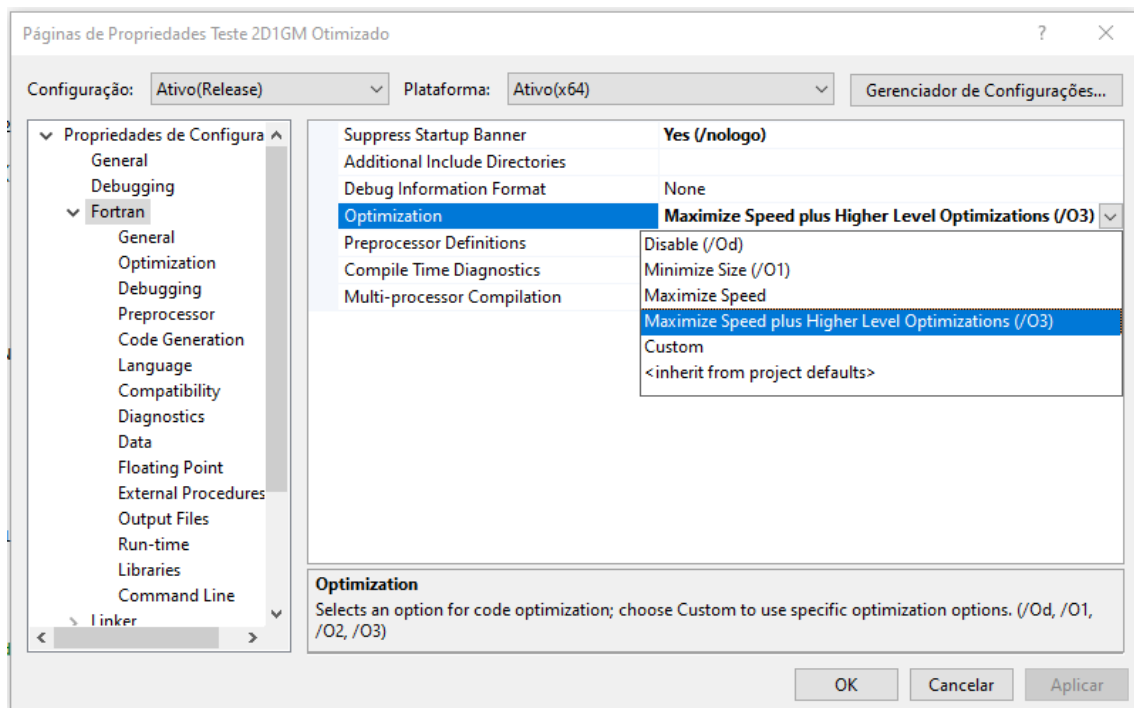


Figura C.1: Geral.

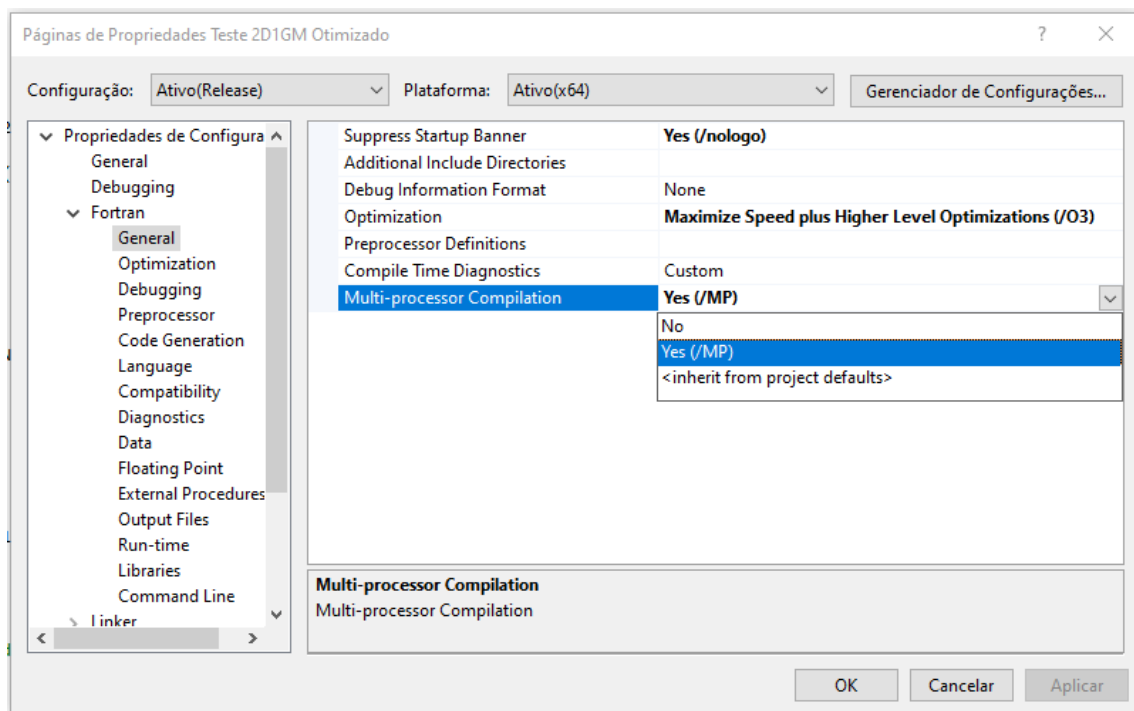


Figura C.2: Multiprocessadores.

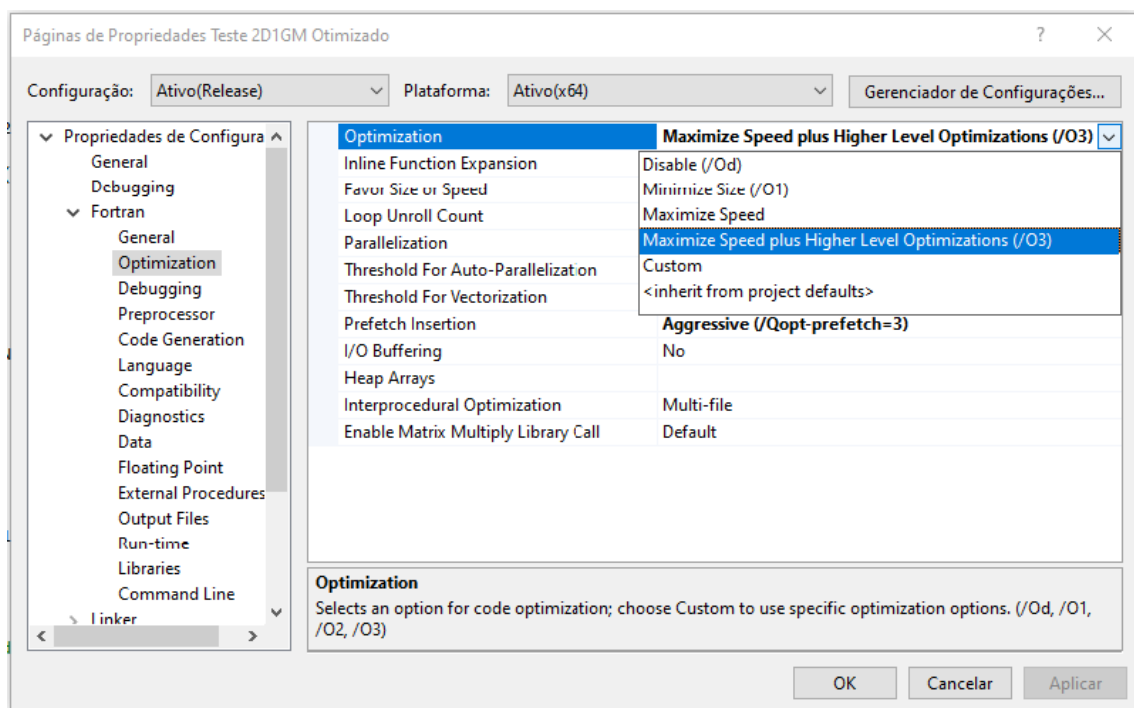


Figura C.3: Nível de Otimização.

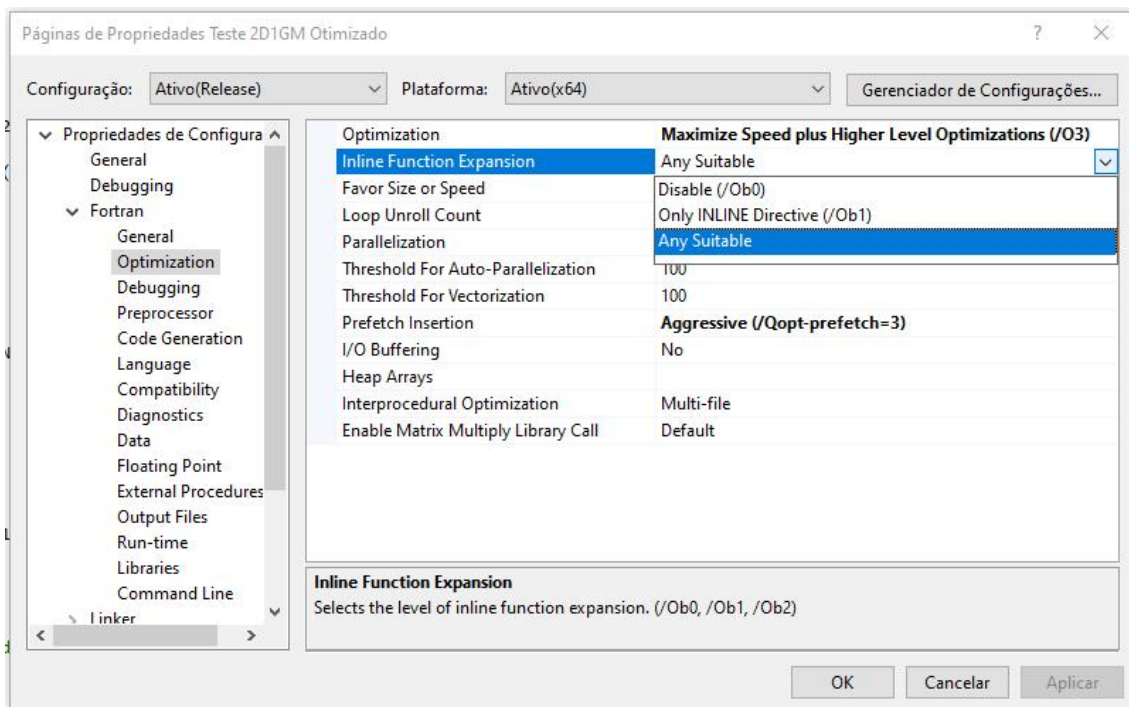


Figura C.4: Função Inline.

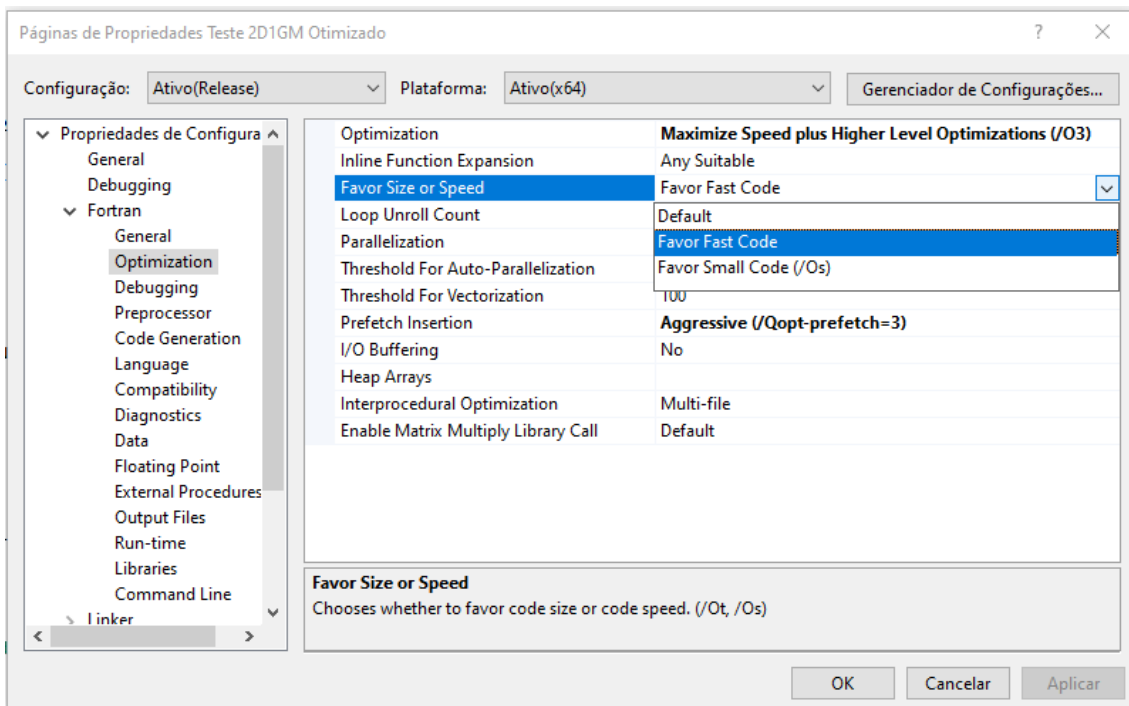


Figura C.5: Tamanho ou Velocidade do Código.

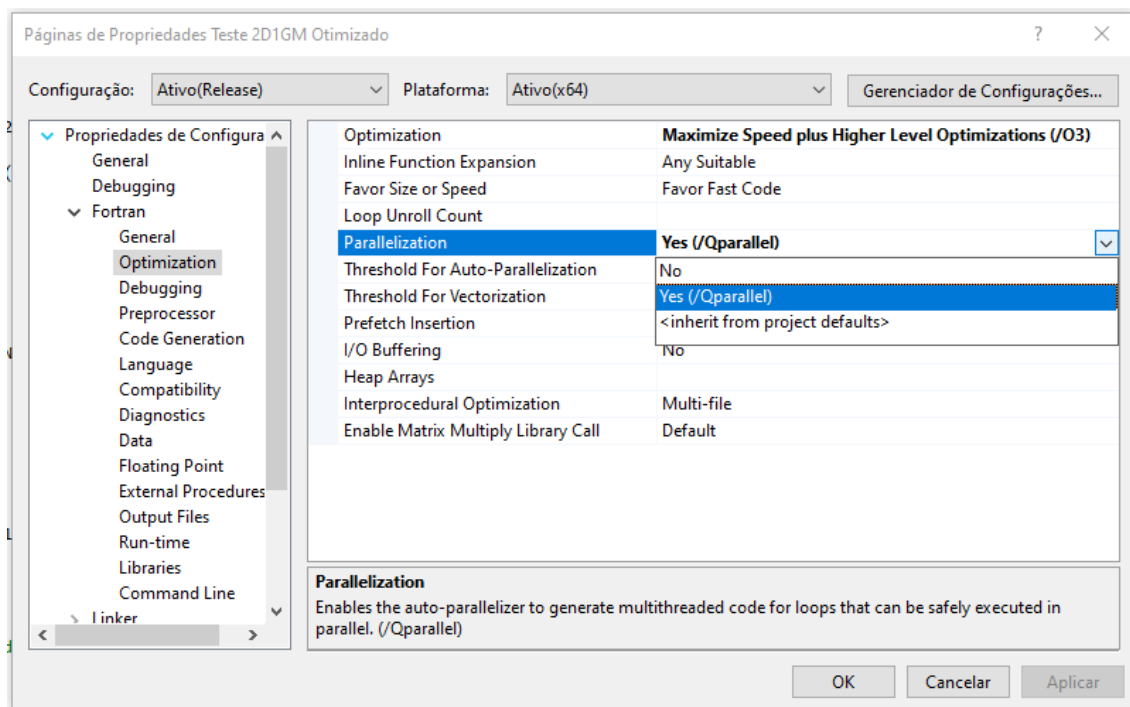


Figura C.6: Paralelização Automática.

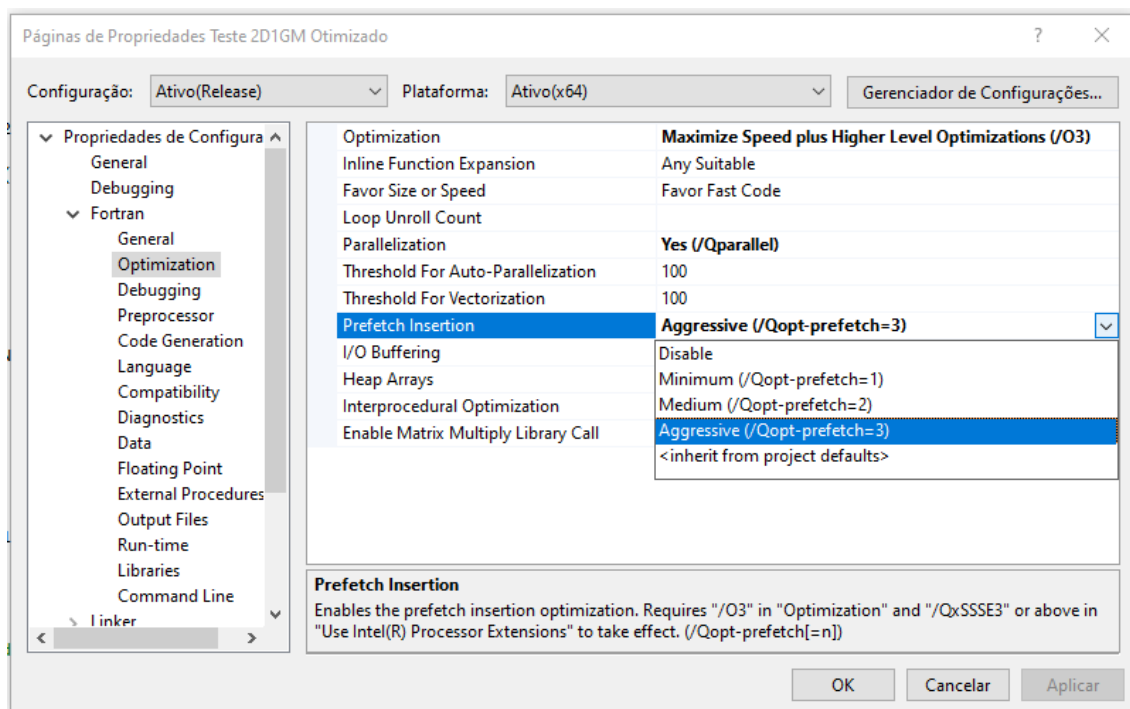


Figura C.7: Pré-carregamento de Cache.

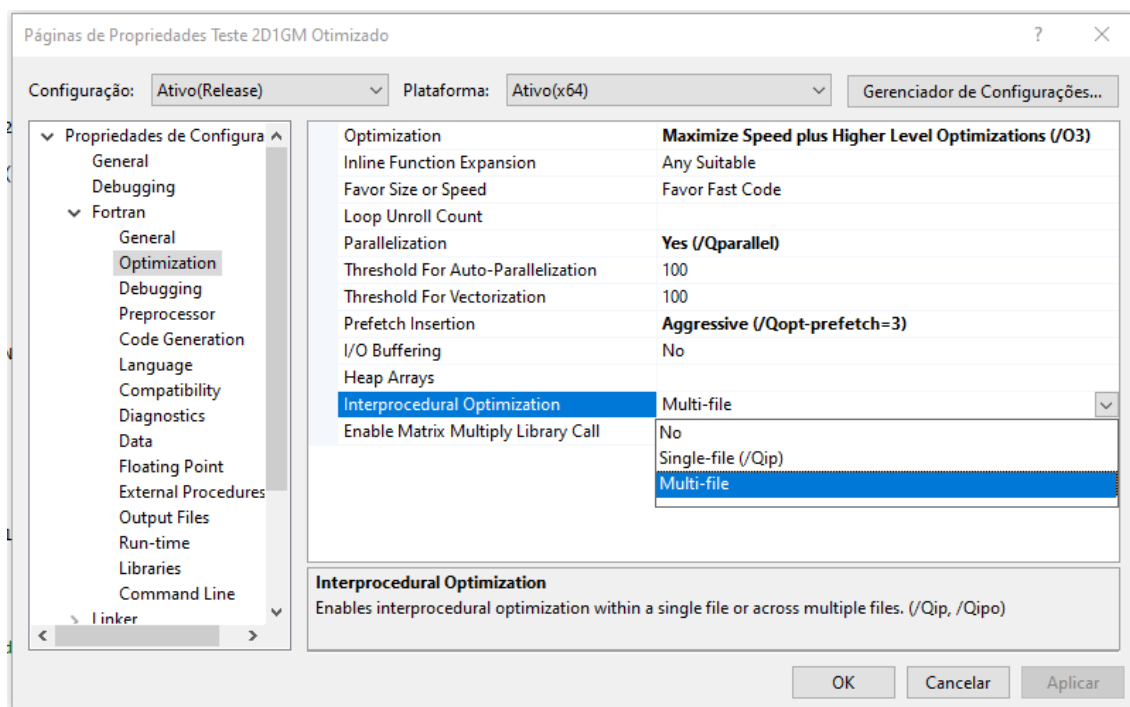


Figura C.8: Otimização Interprocedural.

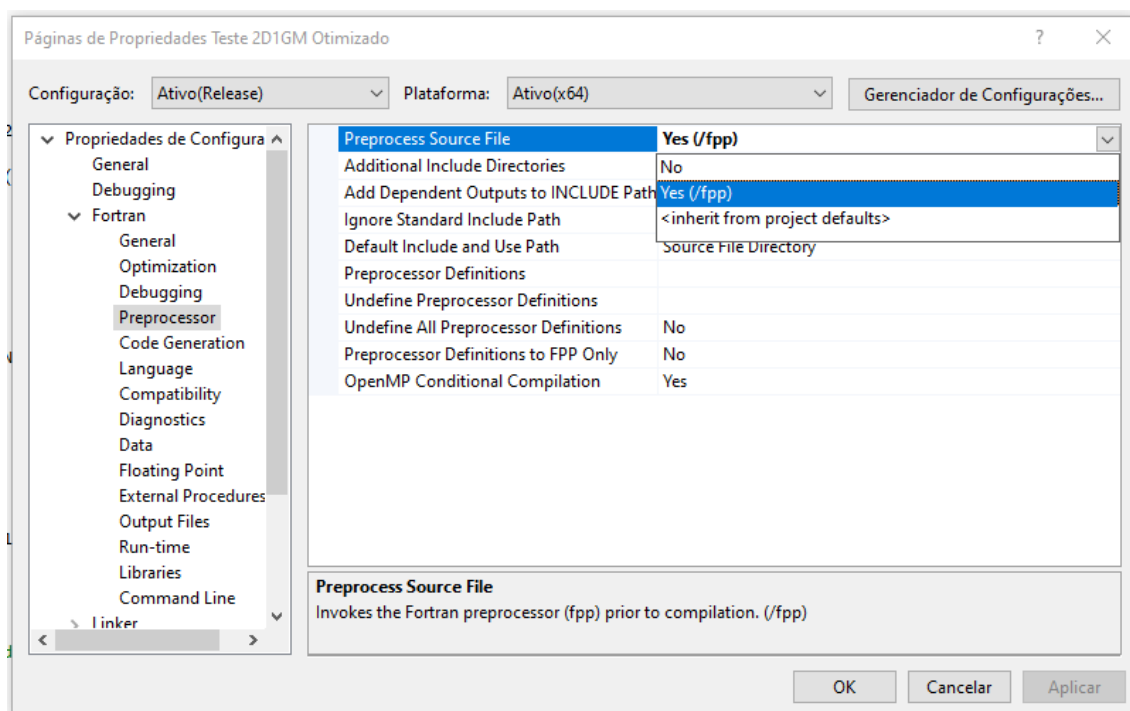


Figura C.9: Pré-processamento.

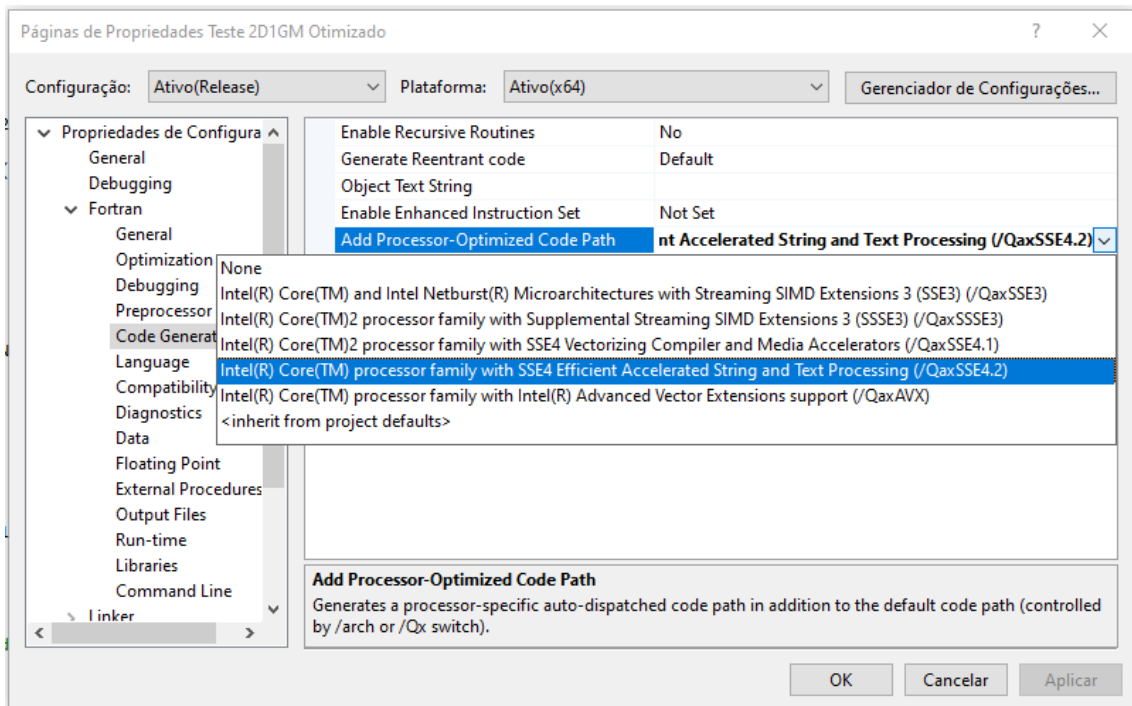


Figura C.10: Geração do código com base nas extensões SIMD.

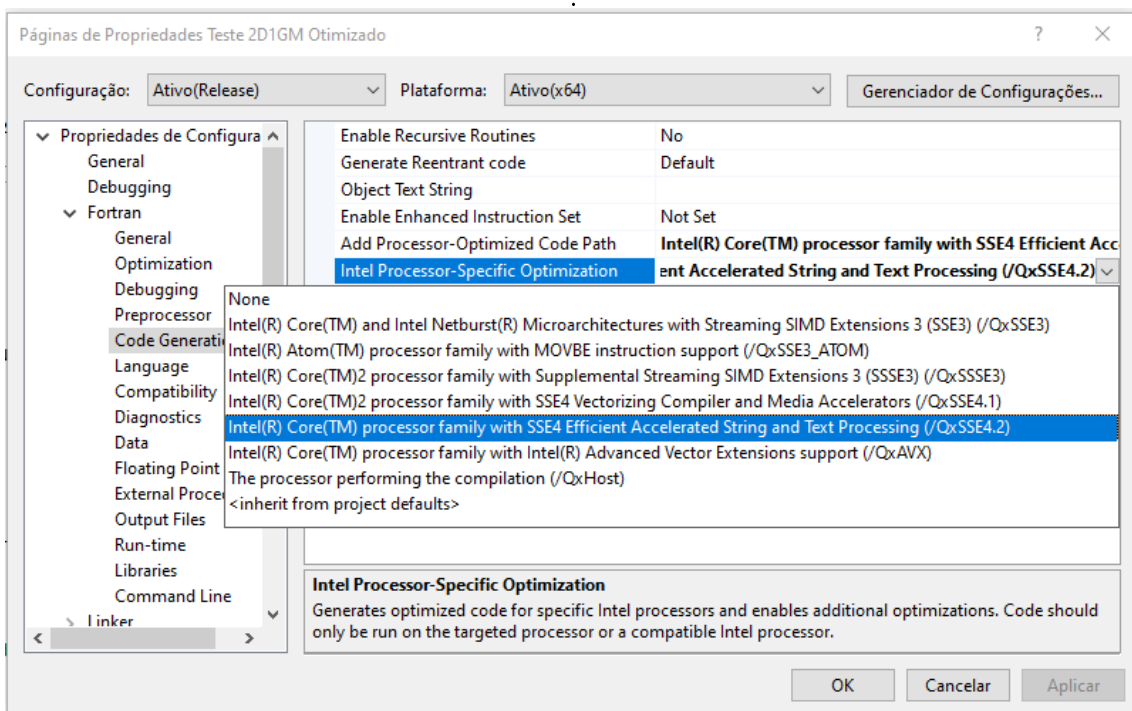


Figura C.11: Geração do código para arquitetura específica.

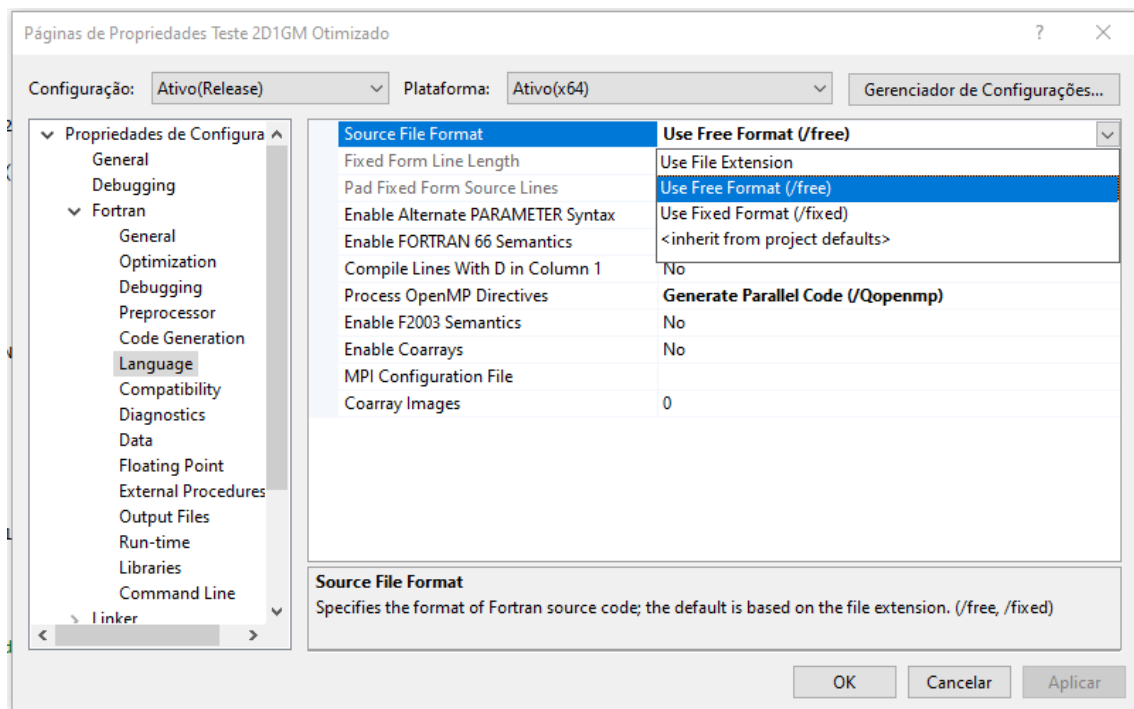


Figura C.12: Formato do Arquivo.

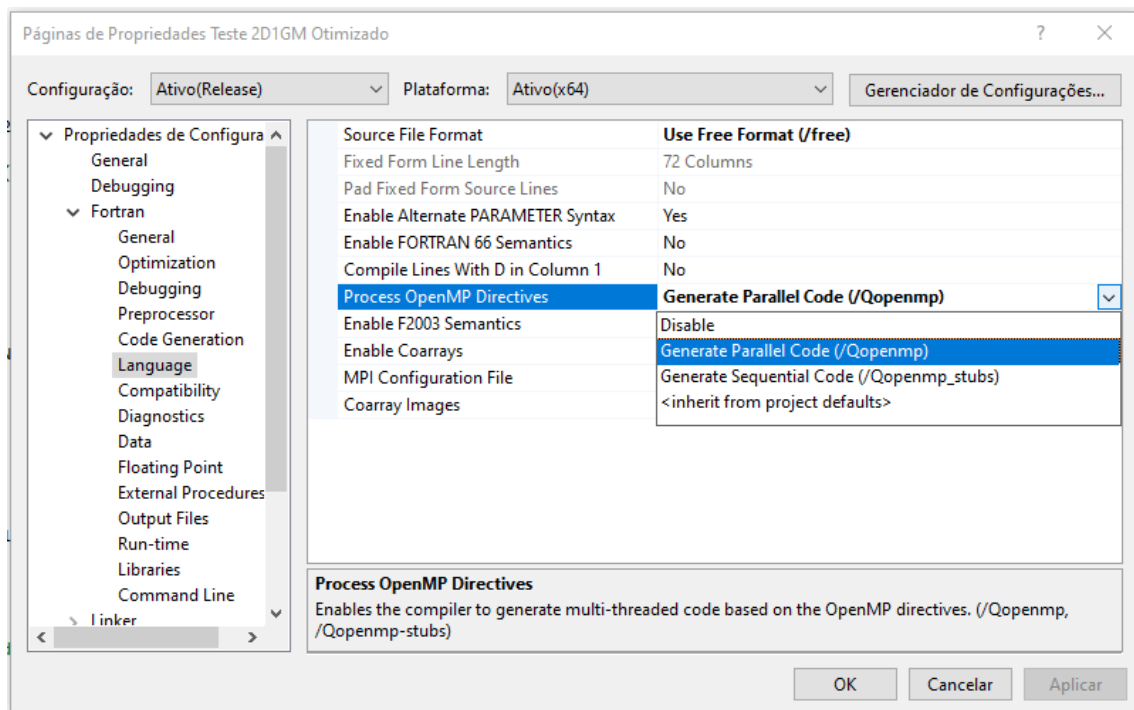


Figura C.13: Diretivas para OpenMP.

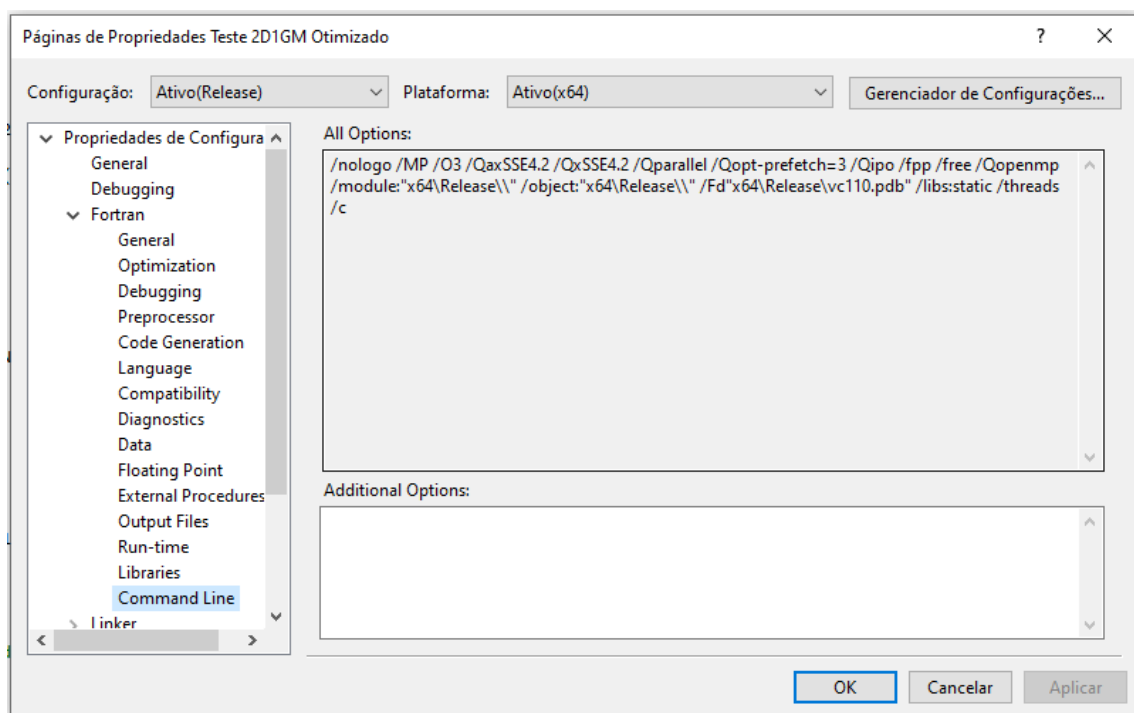


Figura C.14: Linha de comando.

Apêndice D

Queima Média Acumulada

Neste apêndice será demonstrado como os valores de queima acumulada mostrados na Tabela 4.3 foram calculados, assim como a queima média acumulada por EC

$$B(t_l) = B(t_{l-1}) + \Delta B_l, \quad (D.1)$$

onde $B(t_l)$ é o valor da queima num dado instante, $B(t_{l-1})$ o valor de queima no instante anterior.

O incremento de queima a cada instante, ΔB_l , é assim calculado:

$$\Delta B_l = \frac{1}{\rho_{HM}} \bar{P}_{operacao} \Delta t_l, \quad (D.2)$$

e

$$\bar{P}_{operacao} = 10^{-6} f_{carga,l} \bar{P}_{nominal}, \quad (D.3)$$

onde:

$$\bar{P}_{nominal} = \frac{P_{nominal}(em \ MW)}{V_{NucleoAtivo}(em \ cm^3)}, \quad (D.4)$$

$$\Delta t_1 = t_l - t_{l-1}, \quad (D.5)$$

e $B(t_0) = 0$, onde ρ_{HM} é a densidade média de metal pesado¹, em kg/cm^3 ; $P_{nominal}$ é a potência nominal, em MW ; V é o volume do núcleo, em cm^3 ; $f_{carga,l}$ é o fator de carga para cada intervalo; e t_l ; são os instantes de tempo, em dias, para $t = 0, 1, 2, 3, \dots$

A queima média acumulada por EC, $B_{EC}(t_l)$, é dada por:

¹Densidade de metal pesado se refere à massa de material fissil ou fértil por unidade de volume do núcleo do reator.

$$B_{EC}(t_l) = B_{EC}(t_{l-1}) + \bar{P}_{EC}^N \Delta B_l, \quad (\text{D.6})$$

onde, $\bar{P}_{EC}^N$ é a densidade de potência normalizada por EC calculada como:

$$\bar{P}_{EC}^N = (P_{EC}/A_{EC})/(P_{NUCLEO}/A_{NUCLEO}), \quad (\text{D.7})$$

tal que, P_{EC} , P_{NUCLEO} , A_{EC} e A_{NUCLEO} são respectivamente, a potência do EC, a potência do núcleo, a área do EC e a área do núcleo.