



INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Autarquia Associada à Universidade de São Paulo

Modelagem Neutrônica e Termohidráulica de um Reator Modular Rápido Refrigerado à Chumbo-Bismuto Eutético

MATEUS MARIANO GONÇALVES

Dissertação apresentada como parte dos requisitos para obtenção do Grau de Mestre em Tecnologia Nuclear - Reatores

Orientador:
Prof. Dr. Julian Marco Barbosa Shorto

São Paulo

2025

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Autarquia Associada à Universidade de São Paulo

Modelagem Neutrônica e Termohidráulica de um Reator Modular Rápido Refrigerado à Chumbo-Bismuto Eutético

Versão Corrigida
Versão Original Disponível no IPEN

MATEUS MARIANO GONÇALVES

Dissertação apresentada como parte dos requisitos para obtenção do Grau de Mestre em Tecnologia Nuclear - Reatores

Orientador:
Prof. Dr. **Julian Marco Barbosa Shorto**

São Paulo

2025

Autorizo a reprodução e divulgação total ou parcial deste trabalho, para fins de estudo e pesquisa, desde que citada a fonte.

Como citar:

GONÇALVES, M. M. **Modelagem Neutrônica e Termohidráulica de um Reator Modular Rápido Refrigerado à Chumbo-Bismuto Eutético**. 2025. 145 p. Dissertação (Mestrado em Tecnologia Nuclear), Instituto de Pesquisas Energéticas e Nucleares, IPEN-CNEN, São Paulo. Disponível em: <<http://repositorio.ipen.br/>> (data de consulta no formato: dd/mm/aaaa)

Ficha catalográfica elaborada pelo Sistema de geração automática da Biblioteca IPEN, com os dados fornecidos pelo(a) autor(a).

Gonçalves, Mateus Mariano

Modelagem Neutrônica e Termohidráulica de um Reator Modular Rápido Refrigerado à Chumbo-Bismuto Eutético/Mateus Mariano; orientador Julian Marco Barbosa Shorto. São Paulo, 2025. 145 p.

Dissertação (Mestrado) - Programa de Pós-Graduação em Tecnologia Nuclear (Reatores) - Instituto de Pesquisas Energéticas e Nucleares, São Paulo, 2025.

FOLHA DE APROVAÇÃO

Autor: Mateus Mariano Gonçalves

Título: Modelagem Neutrônica e Termohidráulica de um Reator Modular Rápido
Refrigerado à Chumbo-Bismuto Eutético

Dissertação apresentada ao Programa de Pós-Graduação em
Tecnologia Nuclear da Universidade de São Paulo para
obtenção do título de Mestre em Ciências.

Data: 28/08/2025

Banca Examinadora

Prof. Dr.: Julian Marco Barbosa Shorto

Instituição: IPEN-USP

Julgamento: Aprovado

Prof. Dr.: Thiago Carluccio

Instituição: CTMSP – Marinha do Brasil

Julgamento: Aprovado

Prof. Dr.: André Augusto Campagnole dos Santos

Instituição: CDTN

Julgamento: Aprovado

AGRADECIMENTOS

Agradeço à Marinha do Brasil por conceder a oportunidade de me qualificar na área de reatores nucleares, pela qual tenho muito apreço e interesse. Espero que os conhecimentos adquiridos sejam de valia para a instituição.

Agradeço ao meu orientador, Prof. Dr. Julian Marco Barbosa Shorto, pelo tempo e pelos esforços dedicados ao auxílio na escrita desta dissertação.

Agradeço à minha querida Danielle pelo amor, companheirismo e apoio em todos os momentos ao meu lado. Também sou muito grato à Mariana, nosso pequeno anjo que chegou durante a escrita desta dissertação e que Deus enviou para iluminar nossas vidas. Este trabalho é, principalmente e com todo o carinho, dedicado a vocês.

Agradeço também aos meus pais, Pedro Gonçalves e Maria Mariano, e ao meu irmão, Prof. Dr. Vinícius Mariano; o suporte dado por vocês durante toda a minha vida permitiu que eu chegasse até aqui.

Agradeço à banca examinadora desta dissertação pela disponibilidade e pelo empenho em analisar o trabalho de forma crítica, contribuindo assim para torná-lo melhor.

Agradeço aos docentes e servidores do Instituto de Pesquisas Energéticas e Nucleares pelo suporte intelectual e operacional oferecido ao longo destes dois anos.

Por último, sou imensamente grato a Deus por me conceder mais uma conquista da qual tenho muito orgulho.

RESUMO

GONÇALVES, M. M. **Modelagem Neutrônica e Termohidráulica de um Reator Modular Rápido Refrigerado à Chumbo-Bismuto Eutético**. 2025. 145 p. Dissertação (Mestrado em Tecnologia Nuclear), Instituto de Pesquisas Energéticas e Nucleares, IPEN-CNEN, São Paulo.

Esta dissertação propõe uma modelagem multifísica (neutrônica e termohidráulica) de um reator modular pequeno (SMR) rápido refrigerado à liga de chumbo-bismuto eutético (LBE), denominado SPARK-NC e descrito originalmente no artigo de Hashim *et al.*, “*SPARK-NC: A Lead-Bismuth-Cooled Small Modular Fast Reactor with Natural Circulation and Load Following Capabilities*” (2020), que é citado extensivamente ao longo deste trabalho. Tal artigo oferece não só uma miríade de dados geométricos, materiais e parâmetros operacionais que descrevem o núcleo, como também uma série de parâmetros e curvas de desempenho que permitem uma comparação entre o modelo proposto e o modelo original. Com o intuito de validar o modelo multifísico proposto, o trabalho visou reproduzir resultados de referência para comparação com o artigo original, como a evolução temporal do fator de multiplicação efetivo (k_{eff}) e a diferença de massa dos isótopos ^{235}U , ^{238}U e ^{239}Pu devidos à depleção no decurso do ciclo de vida do reator, e os perfis axiais de temperatura na vareta combustível mais quente do núcleo e de potência relativa do núcleo no início do ciclo de vida do reator. Os códigos utilizados para a modelagem foram o OpenMC, que utiliza o método Monte Carlo para a simulação do transporte de partículas neutras e o FiPy, um *solver* de equações diferenciais parciais que emprega o método de volumes finitos. Além da vantagem óbvia de serem gratuitos, ambos são invocados como bibliotecas no ambiente *Python*, o que permite uma integração sem sobressaltos entre os módulos neutrônicos e termohidráulicos, principalmente quanto ao intercâmbio de dados. Para o acoplamento multifísico, implementou-se um mapeamento em que a distribuição de temperatura de uma vareta combustível representativa, obtida com o FiPy através da solução de uma equação de condução-convecção tendo como entrada o perfil axial de densidade de

potência na vareta e a velocidade média do LBE, é estendida às demais varetas combustíveis com base em fatores de escala calculados a partir da razão entre a potência das demais varetas e a potência da vareta representativa. Tanto o perfil axial de potência como os fatores de escala supramencionados são obtidos através da metodologia proposta pelo autor e julgada inédita, que emprega *tallies* descritos por expansões funcionais contínuas e existentes em funções *built-in* do OpenMC. A simulação para o cálculo dos resultados de referência foi realizada por estágios intercalados de ajuste termohidráulico e depleção, onde a distribuição de temperatura em estado estacionário é encontrada para uma dada configuração isotópica do núcleo em uma dada etapa de depleção, com as simulações cobrindo um total de 8.4 anos de operação, que é a faixa temporal de evolução do k_{eff} no artigo original. Os resultados obtidos apresentaram um nível satisfatório de concordância com os dados de referência, não obstante as limitações relacionadas à ausência de informações completas sobre materiais, condições de contorno e diferenças entre os códigos utilizados, indicando que o modelo proposto é capaz de representar com razoável precisão o comportamento neutrônico e termohidráulico do núcleo do SPARK-NC.

Palavras-chave: *Reatores Modulares Pequenos; Reatores Rápidos; Chumbo-Bismuto Eutético; Neutrônica; Termo-Hidráulica; Método Monte Carlo.*

ABSTRACT

GONÇALVES, M. M. **Neutronic and Thermal-Hydraulics Modeling of a Lead-Bismuth Eutectic-Cooled Fast Reactor**. 2025. 145 p. Thesis (Master in Nuclear Engineering), Nuclear and Energy Research Institute, IPEN-CNEN, São Paulo.

This dissertation proposes a multiphysics (neutronic and thermohydraulic) modeling of a fast SMR reactor cooled by a lead-bismuth eutectic alloy (LBE), named SPARK-NC and originally described in the article by Hashim et al., “*SPARK-NC: A Lead-Bismuth-Cooled Small Modular Fast Reactor with Natural Circulation and Load Following Capabilities*” (2020), which is extensively cited throughout this work. The aforementioned article provides not only a wealth of geometrical, material, and operational data describing the reactor core, but also some performance parameters and curves that enable comparison between the proposed model and the original reference. In order to provide a validation for the proposed model, this work aimed to reproduce key reference results, such as the temporal evolution of the effective multiplication factor (k_{eff}), the mass differences of the isotopes ^{235}U , ^{238}U , and ^{239}Pu due to depletion throughout the reactor's lifetime, as well as the axial temperature profiles in the hottest fuel rod and the relative axial power profile of the core at the beginning of the core lifecycle. The codes employed in this study were OpenMC, which uses the Monte Carlo method for simulating the transport of neutral particles (neutrons and photons), and FiPy, a partial differential equations solver based on the finite volume method. Beyond the obvious advantage of being open-source, both are implemented as Python libraries, allowing for seamless integration between the neutronic and thermohydraulic modules—particularly regarding the exchange of data between components. For the multiphysics coupling, a mapping approach was implemented in which the temperature distribution of a representative fuel rod—obtained using FiPy by solving a conduction-convection equation with the rod's axial power density profile and LBE mean velocity as inputs—is extended to all other fuel rods by scaling factors calculated from the ratio between the power generated in the other fuel rods and in the one generated in the representative rod.

Both the aforementioned axial power profile and scaling factors are derived using a methodology proposed by the author and considered original, which employs tallies represented by continuous functional expansions available through built-in OpenMC functions. The simulation used to calculate the reference results was performed through alternating thermohydraulic adjustment and depletion stages, in which the steady-state temperature distribution in each stage is determined for a given isotopic configuration of the core. The complete simulation spans a total of 8.4 years of reactor operation, corresponding to the temporal evolution range of k_{eff} reported in the original article. The results obtained showed satisfactory agreement with the reference data, despite limitations arising from the lack of complete information on material compositions, boundary conditions, and differences between the computational codes employed. These results indicate that the proposed model is capable of representing the neutronic and thermohydraulic behavior of the SPARK-NC core with reasonable accuracy.

Keywords: Small Modular Reactors; Fast Reactors; Lead-Bismuth Eutectic; Neutronics; Thermal-Hydraulics; Monte Carlo Method.

SUMÁRIO

1 - INTRODUÇÃO, REVISÃO BIBLIOGRÁFICA E OBJETIVOS.....	11
1.1 – Visão geral dos SMR.....	11
1.2 - Tipos de SMR e Reatores SMR a Bismuto-Chumbo Eutético (LBE).....	14
1.3 – O SMR refrigerado à Chumbo-Bismuto Eutético SPARK-NC.....	18
1.4 – Objetivos da dissertação.....	19
1.5 – Estrutura da dissertação.....	22
1.6 – Referências do Capítulo.....	22
2 - FUNDAMENTOS TEÓRICOS.....	28
2.1 – A equação de transporte de nêutrons.....	28
2.2 – O problema de autovalores para a obtenção do k_{eff}	35
2.3 – A forma integral da equação de transporte de nêutrons.....	37
2.4 - O método de Monte Carlo para o transporte de nêutrons.....	41
2.5 – A obtenção do k_{eff} pelo método Monte Carlo.....	49
2.6 – Referências do Capítulo.....	50
3 - METODOLOGIA E ANÁLISE.....	51
3.1 – A modelagem do núcleo do SPARK-NC no OpenMC.....	57
3.1.1 – A arquitetura do núcleo do SPARK-NC.....	57
3.1.2 – A modelagem da arquitetura do núcleo do SPARK-NC no OpenMC.....	65
3.1.2.1 – A classe CoreAssemblies	66
3.1.2.2 - A classe CoreMaterials	77
3.1.2.3 - A classe ReactorCore	80
3.2 – O estágio de ajuste termohidráulico.....	88
3.2.1 – A descrição do processo de ajuste termohidráulico.....	88
3.2.1.1 – O cálculo da distribuição de temperatura nos elementos combustíveis.....	90
3.2.2 A modelagem do processo de ajuste termohidráulico.....	112
3.2.2.1 – O cálculo da distribuição de temperatura pela biblioteca FiPy.....	113
3.2.3 – Descrição resumida do estágio de ajuste termohidráulico.....	117
3.3 – O estágio de depleção.....	120
3.3.1 – A descrição do processo de análise de depleção.....	120
3.3.2 – A modelagem da análise de depleção do SPARK-NC no OpenMC.....	124
3.4 – Referências do Capítulo.....	134

4 - RESULTADOS E DISCUSSÃO.....	137
4.1 – Resultados de depleção.....	138
4.2 – Resultados do ajuste termohidráulico.....	141
4.3 – Resultado do perfil axial de potência relativa.....	148
4.3 – Referências do Capítulo.....	148
5 - CONCLUSÃO E DIRECIONAMENTOS FUTUROS.....	149
5.1 – Resumo do trabalho.....	149
5.2 – Cumprimento dos objetivos.....	150
5.3 – Dificuldades e limitações.....	153
5.4 – Direcionamentos Futuros.....	153
5.5 – Considerações Finais.....	154
5.6 – Referências do Capítulo.....	155

CAPÍTULO 1

INTRODUÇÃO, REVISÃO BIBLIOGRÁFICA E OBJETIVOS

Este capítulo oferece uma introdução no contexto do trabalho realizado nesta dissertação, citando algumas referências bibliográficas acerca do assunto. Primeiramente, é dada uma visão geral sobre os reatores SMR (*Small Modular Reactors*, ou Reatores Modulares Pequenos), os tipos de SMR e, mais especificamente, aqueles refrigerados à liga de chumbo-bismuto eutético (LBE, *Lead-Bismuth Eutectic*). Então, o SMR refrigerado à LBE SPARK-NC proposto em [1] é descrito com suas especificidades, permitindo enunciar o objetivo central da dissertação, que é a modelagem neutrônica e termohidráulica do SPARK-NC e a comparação de alguns resultados com aqueles obtidos na referência original [1]. Por fim, uma descrição da estrutura da dissertação é mostrada.

1.1 – Visão geral dos SMR

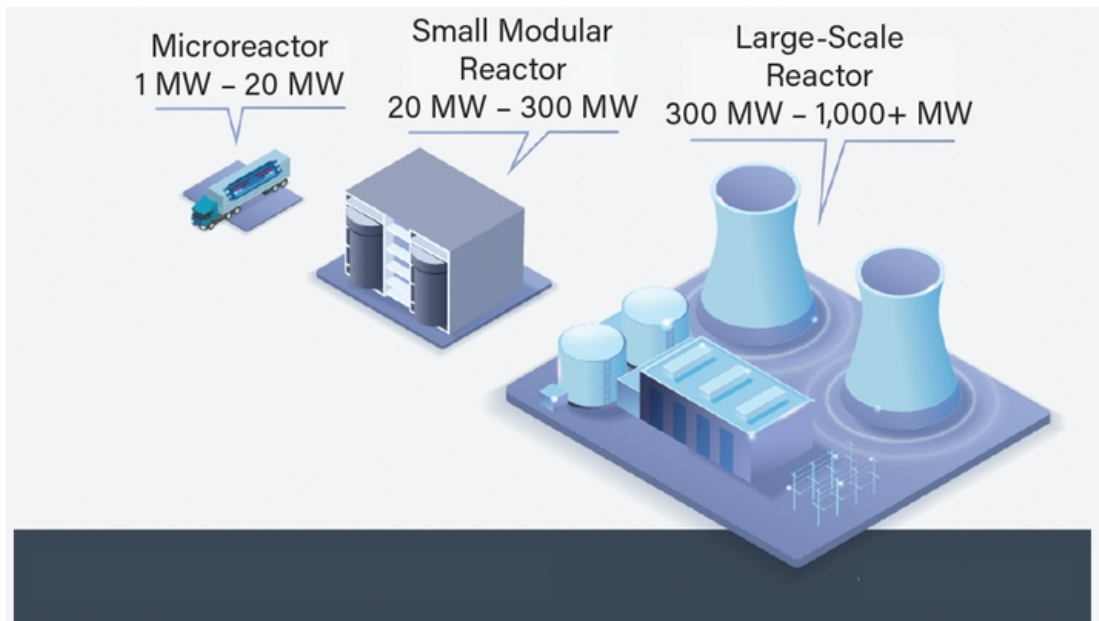
Os SMR (*Small Modular Reactors*, ou Pequenos Reatores Modulares) são uma classe de plantas nucleares cuja proposta é relativamente recente, dada parcialmente em resposta aos desafios impostos pelas dimensões e complexidades intrínsecas dos reatores atuais, que acabam por limitar o seu amplo emprego, mormente devido às questões relacionadas à segurança e ao custo de construção e comissionamento. Esses desafios técnicos e financeiros, documentados em estudos recentes, ressaltam a necessidade de alternativas mais flexíveis e escaláveis para a geração de energia nuclear [2].

Os SMR são reatores projetados para possuírem tamanho e capacidade de geração reduzidos (ver Figura 1.1), permitindo serem produzidos em escala em uma única planta fabril e movimentados para diferentes locais para serem operados, o que em tese possibilita a diluição dos custos do fabricante a médio e longo prazo devido à manufatura centralizada e padronizada, reduzido tempo de construção, e à necessidade de um único processo de licenciamento por projeto [3]. A modularidade desses reatores tem sido amplamente estudada por instituições internacionais, que enfatizam a redução dos riscos associados ao licenciamento específico de cada unidade [4].

Sob a ótica do cliente, as possíveis vantagens dos SMR incluem o fornecimento energético escalável, permitindo a inclusão de novos módulos à rede conforme a demanda aumenta; o custo reduzido, já que a aquisição de um SMR pode ser economicamente vantajosa em relação a um projeto maior, por exigir uma infraestrutura menos robusta; e a flexibilidade, decorrente de sua característica modular, que permite adaptá-los tanto a instalações convencionais quanto a locais remotos com infraestrutura limitada [5]. Essas vantagens são especialmente relevantes em regiões

onde a demanda energética cresce de forma gradual ou onde há limitações logísticas para a instalação de grandes usinas nucleares.

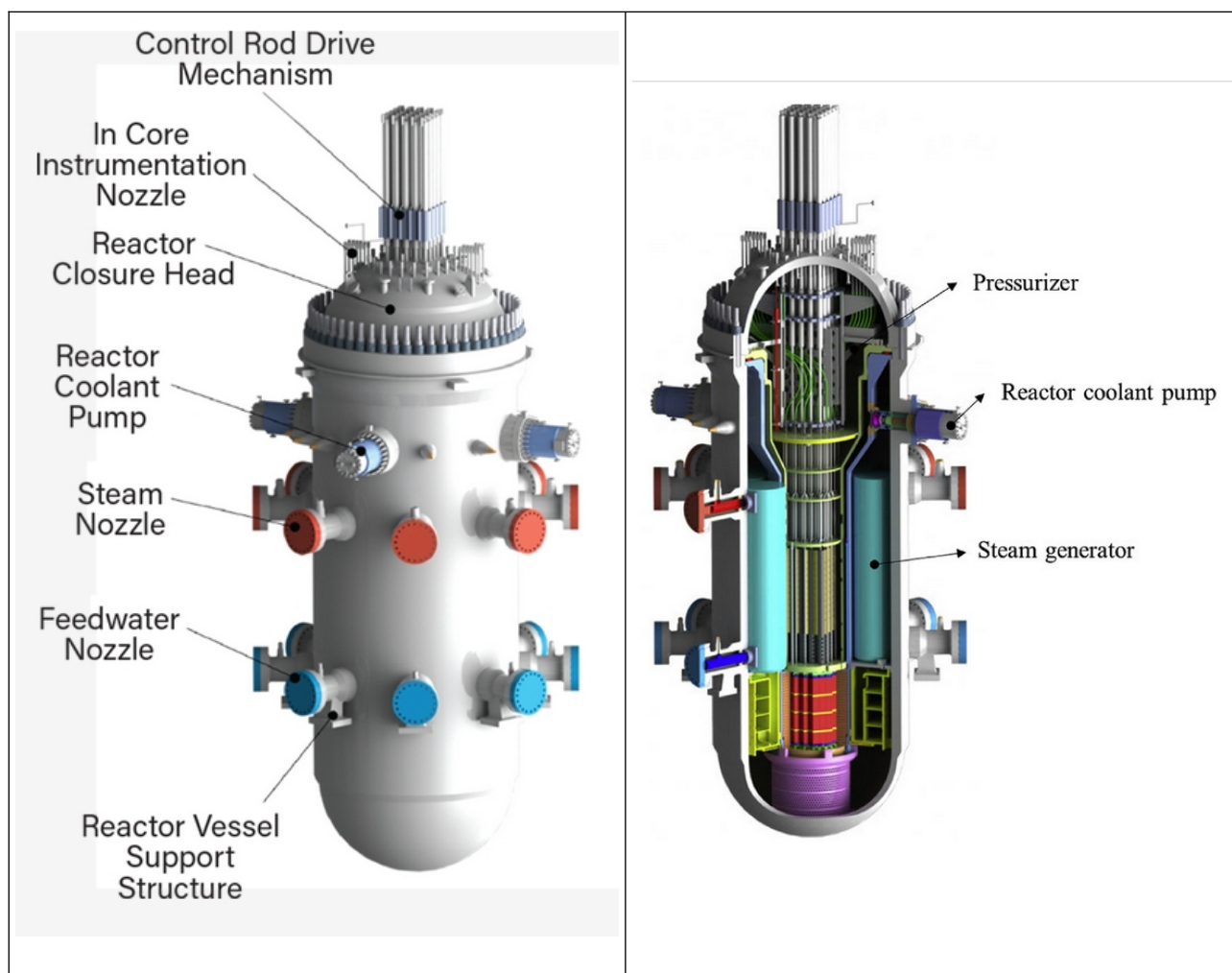
Figura 1.1 –Classificação de reatores conforme a faixa de potência gerada. Os SMR situam-se entre os microrreatores (também chamados de vSMR, *very Small Modular Reactors*) e os reatores convencionais de larga escala. Retirado de [6].



Em adição às vantagens já citadas, projetos de SMR oferecem benefícios significativos quanto à segurança da operação. De forma geral, esses reatores apresentam um projeto mais simples em comparação com os reatores convencionais e frequentemente incorporam características inerentes dos reatores de quarta geração, bem como mecanismos de segurança passiva típicos dos reatores de terceira geração, por exemplo a circulação e convecção natural, ação por gravidade e auto-pressurização [3]. Tais medidas de segurança contribuem para uma margem operacional mais elevada, reduzindo substancialmente o potencial de emissões não seguras de radioatividade em caso de acidente e minimizando os riscos ambientais e à saúde pública. Uma das diretrizes adotadas em diversos projetos SMR, que é a integração dos componentes principais, como o núcleo e auxiliares, dentro de um único vaso, fortalece a capacidade de contenção de emissões radioativas acidentais e reduz a vulnerabilidade a falhas por rupturas de tubulação (ver Figura 1.2), que oferece uma ameaça significativa de acidentes em reatores convencionais [7]. Além disso, propostas de reatores SMR frequentemente utilizam combustíveis não convencionais que permitem um maior *burnup* e ciclos de troca de combustível mais dilatados, diminuindo a frequência de reabastecimento e, conseqüentemente, os riscos de acidentes e de proliferação nuclear. Essa abordagem inovadora

também contribui para uma gestão eficaz da utilização do combustível, promovendo uma melhor gestão dos resíduos nucleares e aumentando a eficiência energética.

Figura 1.2 – Esquemático do reator SMR SMART (*System-integrated Modular Advanced Reactor*) desenvolvido na Coreia do Sul, com o corte mostrando a sua estrutura interna. Nota-se que elementos como as bombas de refrigerante (*Reactor Coolant Pump*), geradores de vapor (*Steam Generator*) e pressurizador (*Pressurizer*) estão alojados dentro do vaso de pressão ou diretamente acoplados ao mesmo, o que mostra que o reator encontra-se alinhado com o princípio de integração de componentes adotado em diversos projetos SMR. Retirado de [6].



Apesar das críticas, principalmente no que se refere às dificuldades para atingir economia de escala e à produção em massa de pequenas usinas, os SMR têm conquistado crescente apoio da academia, de agências regulatórias e de governos locais. Em 2024, estima-se que existam mais de oitenta projetos de SMR em andamento em 19 países [4] (incluindo o CAREM, sob construção na Argentina [8]), com protótipos já em operação na Rússia (os dois SMR à água pressurizada KLT-40S a bordo da usina flutuante *Akademik Lomonosov*) e China (o SMR a leito de seixos refrigerado a gás HTR-PM) [9], demonstrando o potencial estratégico dessa tecnologia para o futuro da energia nuclear. De maneira complementar, as pesquisas e os investimentos em todo o mundo reforçam a

viabilidade técnica e econômica dos SMR, destacando-os como uma solução promissora para diversificar a matriz energética e reduzir a dependência de combustíveis fósseis. Pesquisas recentes enfatizam que a disseminação em larga escala desses reatores pode acelerar a transição energética e contribuir para o cumprimento de metas globais de redução de emissões de carbono [2], além de auxiliar na demanda massiva por fontes distribuídas e não-intermitentes de energia para os futuros centros de processamento baseados em IA (inteligência artificial).

A capacidade promissora dos SMR de serem integrados a diversas aplicações, como a cogeração de calor para processos industriais, produção de hidrogênio por eletrólise, dessalinização de água e até mesmo aplicações em propulsão naval, intensifica seu futuro potencial de mercado. Essas múltiplas aplicações reforçam o papel dos SMR como um elemento chave na diversificação dos sistemas de energia modernos, promovendo segurança energética e inovação tecnológica.

1.2 - Tipos de SMR e Reatores SMR a Bismuto-Chumbo Eutético (LBE)

Os projetos propostos de SMR são diversos e contemplados por uma literatura extensa, sendo alguns simplificações de projetos de reatores convencionais, enquanto outros apresentam propostas radicalmente diferentes. Dentre os projetos convencionais, podem ser observados aqueles à água pressurizada e fervente (PWR, *Pressurized Water Reactor* e BWR, *Boiling Water Reactor*), reatores refrigerados a gás (GCR, *Gas-Cooled Reactors*), reatores a sal fundido (MSR – *Molten Salt Reactors*) e reatores rápidos refrigerados a metal líquido (LMFR, *Liquid Metal Fast Reactors*).

Da mesma maneira que os PWR são os reatores convencionais de larga escala com o maior número de plantas no planeta [10], os SMR refrigerados à água pressurizada são projetos de reatores modulares mais comuns, provavelmente devido ao fato de serem um projeto mais antigo e já consagrado na indústria, apesar da existência de componentes de grande escala em relação às dimensões do vaso (pressurizador, gerador de vapor, etc.), cuja integração no interior do vaso de pressão pode oferecer desafios significativos em um SMR desse tipo. Dentre os projetos mais significativos de SMR PWR, destacam-se O ACP-100 chinês de 385 MWt [11], que já se encontra em construção (a sua estrutura interna principal foi concluída em 2023 [12]), os dois já supramencionados KLT-40S a bordo da usina flutuante *Akademik Lomonosov* (ver Figura 1.3), com operação iniciada em 2020 [13], o também já mencionado CAREM argentino de 25 MWe [8], também em construção, cujo projeto contempla circulação natural, os RITM-200S e RITM-200N, sob construção na Rússia, que serão utilizados em terra e embarcados em navios quebra-gelo [14]. O reator SMART, na Coreia, e o VOYGR, da empresa *NuScale*, nos Estados Unidos da América, que é composto por vários pequenos reatores denominados *NuScale Power Modules* (NPM), são exemplos de SMR PWR que já foram licenciados ou parcialmente licenciados pelas autoridades

nucleares de seus respectivos países [15],[16]. Por fim, o SMR à água pressurizada da *Rolls-Royce*, no Reino Unido [17], e o SMR-300 da *Holtec International*, nos Estados Unidos [18], são modelos PWR que já estão com projeto concluído e em processo de licenciamento nos Estados Unidos e na Grã-Bretanha para construção futura. A *Westinghouse Electric Company* iniciou em 2023 um projeto de SMR PWR, o AP300, escalado a partir do já existente AP1000 [19].

Figura 1.3 - SMR russo KLT-40S sendo embarcado à bordo do navio *Akademik Lomonosov*. Retirado do sítio globalsecurity.org em 2 abr. 2025.



Os projetos de SMR BWR são menos numerosos, sendo o de maior destaque o BWRX-300, em projeto pelo consórcio Nipo-Americano *GE Hitachi Nuclear Energy*. Segundo os projetistas, o projeto contempla medidas de segurança passiva que prescindem de energia externa e/ou ação do operador até nas circunstâncias acidentais mais extremas [20].

Dentre os reatores SMR refrigerados a gás, menciona-se o chinês HTR-PM, que é um reator a leito de seixos refrigerado a gás em alta temperatura HTGR (*High-Temperature Gas-Cooled Reactor*, ver Figura 1.4), que já está em operação na província chinesa de *Shandong*, próximo à península coreana, desde o fim de 2023, com dois reatores de 250 MW térmicos conectados a uma única turbina, provendo uma geração total de 210 MW elétricos [21].

Os reatores SMR refrigerado a sal fundido atualmente contemplam, dentre outros, o TMSR-LF1, que foi projetado pela CNNC (*China National Nuclear Corporation*) e já opera como uma planta nuclear piloto produzindo 2 MW térmicos no noroeste da China. O combustível líquido utilizado possui uma mistura de fluoretos de Urânio enriquecido a 19.75%, Zircônio e Tório, dissolvidos em um sal fundido de Fluoreto de Lítio e Fluoreto de Berílio. A criticalidade da planta

foi atingida em outubro de 2023, com a potência plena tendo sido atingida em junho de 2024 [22]. O ISMR é um reator a sal fundido (fluoretos de sódio, lítio e berílio) com núcleo integrado, que está em fase de desenvolvimento pela empresa canadense *Terrestrial Energy*, tendo completado seu pré-licenciamento junto às autoridades nucleares canadenses [23].

Figura 1.4 – Carregamento de combustível no SMR HTR-PM. Retirado do sítio *nucnet.org*, acessado em 2 abr. 2025.



Dentre os SMR rápidos a metal líquido (LMFR), possuem destaque o BREST-OD-300, refrigerado a chumbo, com capacidade de produção de 300 MWe e utilizando combustível misto de urânio e plutônio, que já se encontra em construção em Tomsk, na Rússia [24], e o SEALER, também refrigerado a chumbo e que ainda se encontra na fase de projeto na Suécia [25] .

Também na categoria dos LMFR, os reatores refrigerados a liga de chumbo-bismuto eutético (LBE, Lead-Bismuth Eutectic) apresentam uma série de benefícios em relação a outros projetos de reatores rápidos refrigerados a metal líquido. Entre esses benefícios, destaca-se o baixo ponto de fusão e o alto ponto de ebulição da liga, que permitem ao refrigerante permanecer em fase líquida durante toda a faixa operacional e mesmo em condições incidentais de superaquecimento, o que contribui para maior eficiência térmica e elimina a necessidade de pressurização durante a operação normal [26]. Essa característica também amplia a margem de segurança passiva do projeto, que é um aspecto essencial na redução dos riscos operacionais.

A inércia química elevada do LBE – em contraste com o sódio líquido utilizado em alguns LMFR que pode reagir de maneira explosiva com o ar ou a água – é um ponto crucial para a

segurança, pois minimiza o risco de reações indesejadas em caso de falhas e vazamentos em circuitos de refrigeração [27]. Outro benefício importante é a blindagem radiológica inerente, já que o LBE possui uma seção de choque significativa para a absorção de radiação gama, o que contribui para a proteção dos operadores e da população ao redor da planta. O sódio líquido, por sua vez, forma uma grande quantidade do isótopo ^{24}Na via absorção de nêutrons, sendo o mesmo um emissor gama de alta intensidade.

O elevado coeficiente de expansão térmica do LBE torna possível a implementação de medidas passivas baseadas em circulação natural, promovendo uma remoção eficiente do calor sem a necessidade de sistemas ativos de bombeamento. Tal característica, apesar das diversas vantagens oferecidas, ainda é objeto de pesquisas extensivas que procuram mitigar os seus efeitos colaterais indesejados, como um menor fluxo de massa e uma maior instabilidade de fluxo, quando comparado com sistemas de circulação forçada [28].

Apesar das vantagens apresentadas, os reatores LBE possuem desafios técnicos a serem superados, como a tendência à corrosão dos materiais estruturais e de revestimento devido à alta solubilidade da liga em altas temperaturas. Outro ponto crítico é a formação do Polônio-210, um poderoso alfa-emissor resultante da captura de nêutrons pelo Bismuto, que exige medidas rigorosas de controle e manejo do material radioativo [29]. Esses aspectos têm estimulado pesquisas para o desenvolvimento de ligas aprimoradas e técnicas de revestimento que mitiguem os efeitos corrosivos, conforme demonstrado em diversos estudos de materiais avançados [26].

A antiga União Soviética foi pioneira na tecnologia LBE ao implementar os primeiros reatores térmicos refrigerados a Chumbo-Bismuto Eutético a bordo dos submarinos classe *Alfa*, comissionados no início dos anos 70 [30]. A Rússia ainda mantém centros de pesquisa com reconhecida *expertise* em engenharia de reatores refrigerados à LBE, tendo projetado um SMR rápido denominado SVBR-75/100, que utiliza um combustível híbrido de óxidos de plutônio e urânio, com resfriamento do núcleo a chumbo-bismuto líquido circulando por um circuito primário integrado ao vaso de pressão [31].

Diversos outros projetos de SMR que adotam a tecnologia LBE foram propostos ou executados nos últimos 20 anos. Na Coreia, foram propostos o PASCAR, um reator compacto de larga autonomia que utiliza LBE como refrigerante e cujo projeto é orientado para oferecer resistência à proliferação e tolerância a acidentes [32], e o Micro-URANUS, uma adaptação moderna do reator convencional URANUS que incorpora a tecnologia LBE [33]. No Japão, há o projeto do LSPR (*LBE-Cooled Long-Life Safe Simple Small Portable Proliferation-Resistant Reactor*) que visa alcançar um ciclo de troca de combustível de 30 anos [34], enquanto na China os projetos SUMMER (*SUstainable Modular Mobile Enhanced Reactor*) [35] e SPARK [36]

evidenciam a viabilidade de reatores transportáveis com alta autonomia para operação em áreas remotas. Mais recentemente, também na China, o NCLFR-Oil (*Natural Circulation Lead-bismuth Fast Reactor for Oil extraction*) foi concebido para fornecer energia em poços de extração de petróleo, demonstrando a versatilidade dos SMR refrigerados a LBE em diversas aplicações [37].

1.3 – O SMR refrigerado à Chumbo-Bismuto Eutético SPARK-NC

O SMR refrigerado à LBE SPARK-NC [1], proposto em 2020 por pesquisadores da Universidade *Xi'an Jiaotong* na China, apresenta uma evolução em relação ao reator SMR SPARK [36], mencionado na Seção 1.2, ao prescindir do sistema de bombas do primário, incorporando um sistema avançado de circulação natural. Tal avanço é benéfico, pois simplifica a infraestrutura geral da planta ao remover o sistema intermediário ativo de transporte de calor, que é composto de bombas cuja tecnologia, projeto e manutenção são especialmente complexos em sistemas LBE, dada a alta densidade da liga e a já mencionada tendência à corrosão e erosão dos materiais componentes do reator. Adicionalmente, o SPARK-NC possui capacidade de seguimento de carga, permitindo que a planta ajuste sua geração de potência à flutuação da demanda.

As características físicas, geométricas e de operação do SPARK-NC são descritas de maneira detalhada em [1], sendo também dado um panorama de sua arquitetura na Seção 3.1. É um reator rápido operando com varetas combustíveis de óxido de urânio (UO_2) enriquecido até 19.75% como combustível com ciclo de oito anos em potência máxima, possuindo uma faixa de geração que pode ser ajustada entre 5 MWe e 10 MWe de acordo com a demanda. O projeto lida com os efeitos danosos da corrosão/erosão pela LBE em altas temperaturas através de uma estratégia bivalente, tanto utilizando o aço T91, uma liga especialmente resistente à corrosão LBE, quanto mantendo a temperatura de pico no revestimento do combustível dentro de valores aceitáveis.

O projeto do núcleo é fornecido em [1] com um nível elevado de detalhamento, possuindo 8 tipos de elementos: três níveis concêntricos de elementos combustíveis, elementos de blindagem e de refletores, elementos de controle de reatividade primários e secundários e elementos de SCRAM, com sua posição relativa dada na configuração radial e axial do núcleo. A geometria do núcleo é completamente fornecida, com dados diversos como número de elementos com o número de varetas para cada elemento, o *pitch*, a altura e raio de cada componente de cada elemento (refletores, absorvedores, combustível, *plenum* para acúmulo de gases, etc.), espessura do revestimento e do *gap*, além da especificação dos materiais envolvidos. Por fim, são listados parâmetros nominais do projeto como a potência térmica do núcleo, o inventário de metal pesado no início do ciclo de vida do reator, e a temperatura de entrada, vazão mássica e a velocidade média do fluido refrigerante.

1.4 – Objetivos da dissertação

No artigo original que descreve o SPARK-NC [1], além da descrição minuciosa das características geométricas e materiais do núcleo mencionada na Seção 1.3, os autores obtiveram diversos parâmetros de performance para o projeto do núcleo, como a evolução do fator de multiplicação efetivo e a variação em massa de isótopos ao longo do ciclo de vida do reator, curvas de pico de potência relativos do núcleo e coeficientes de reatividade, além de parâmetros termohidráulicos como o perfil de temperatura da vareta mais quente no núcleo. Tais cálculos foram originalmente feitos com um auxílio de um *software* de simulação determinística, que é proprietário da Universidade *Xi'an Jiaotong*, na China, denominado SARAX (*System for Advanced Reactor Analysis at Xi'an Jiaotong University*) [38]. Tal sistema é pouco conhecido e disseminado: não se tem acesso ao código, sendo as referências aos trabalhos que o utilizam, e ao sistema propriamente dito, relativamente escassas.

Por outro lado, a miríade de dados oferecidos em [1] torna possível a modelagem geométrica e de materiais do núcleo do SPARK-NC em um código de análise neutrônica da escolha do usuário, que pode então compará-los com resultados dos cálculos realizados em [1]. Deste modo, o primeiro objetivo desta dissertação foi a realização da modelagem do núcleo do SPARK-NC com a utilização da biblioteca do *Python* denominada **OpenMC** desenvolvido pelo Grupo de Física de Reatores Computacional do MIT (*Massachusetts Institute of Technology*) [39], que oferece um conjunto amplo e compreensivo de funções para a modelagem e análise neutrônica utilizando o método Monte Carlo (ver Seção 2.4), ainda possuindo a vantagem adicional de ser um código livre e completamente gratuito, em contraste a outros códigos mais comumente utilizados, como o MCNP.

Conforme discutido em [1], o SARAX possui um módulo termohidráulico com o qual as curvas de distribuição axial de temperatura na região da vareta mais quente do SPARK-NC foram obtidas através de um cálculo de subcanal acoplado ao módulo neutrônico. Entretanto, a descrição detalhada de como tal módulo é utilizado para o cálculo dessas curvas, incluindo as equações fundamentais do modelo, valores de parâmetros termohidráulicos, hipóteses empregadas e condições de contorno, não é fornecida em [1]. Visto que uma simulação multifísica (neutrônica e termohidráulica) do núcleo é essencial para que se tenha uma comparação consistente com os resultados observados em [1], o segundo objetivo deste trabalho foi o desenvolvimento de uma metodologia para a obtenção de um módulo termohidráulico responsável pelo cálculo da distribuição de temperatura no núcleo do SPARK-NC, além de um algoritmo para o acoplamento

desse módulo com o OpenMC de modo a permitir o intercâmbio de parâmetros neutrônicos e térmicos entre as partes.

O fato do código OpenMC estar no ambiente *Python* torna possível o seu acoplamento de maneira natural à uma das diversas bibliotecas gratuitas de análise térmica e de fluidos também presentes em tal ambiente. Isso permite que tarefas como troca de dados entre módulos neutrônicos e termohidráulicos, que podem constituir tarefas não-triviais em códigos de linguagem exclusiva como o MCNP, possam ser executadas sem esforço adicional, encapsulando o processamento e pós-processamento multifísico dessas duas componentes em um único programa. Assim, para o cálculo numérico da distribuição de temperatura na vareta representativa o autor escolheu o módulo do *Python* **FiPy** [40] que soluciona uma classe extensa de equações diferenciais parciais por volumes finitos. Devido às propriedades térmicas peculiares de fluidos refrigerantes baseados em metais líquidos, assumiu-se para a descrição dos fenômenos termohidráulicos na vareta representativa um modelo relativamente simples de difusão-convecção (Ver Seção 3.2).

Para se obter a supramencionada distribuição de temperatura do núcleo do modelo do SPARK-NC, desenvolveu-se um algoritmo em que calcula-se, através do modelo de difusão-convecção mencionado acima, a distribuição de temperatura da região associada a apenas uma vareta combustível representativa, que é elegida como a vareta que apresenta a maior produção calorífica no núcleo, com a distribuição de temperatura das regiões pertencentes às outras varetas combustíveis sendo então determinadas por um mapeamento que leva em consideração a produção calorífica relativa entre a vareta representativa e as varetas restantes. Tal algoritmo, descrito em detalhes na Seção 3.2, emprega a distribuição contínua de densidade linear axial de produção calorífica na região ativa do núcleo como uma expansão em polinômios ortogonais de Legendre (tal expansão é obtida através de funções existentes no OpenMC que fornecem expansões funcionais contínuas de *tallies* [41]), além dos *tallies* de produção calorífica total em cada vareta combustível. O autor presume, dada a inexistência de referências similares, que tal abordagem, utilizando esses *tallies* específicos para o cálculo da distribuição térmica no núcleo, seja inédita.

Os dois objetivos propostos acima, que são a obtenção de um modelo do núcleo no SPARK-NC no OpenMC e a criação de uma metodologia de acoplamento neutrônico/termohidráulico para esse modelo, devem oferecer uma forma de validação para que o cumprimento desses objetivos possa ser mensurado. Desta maneira, um objetivo subsidiário deste trabalho foi a comparação de algumas curvas e parâmetros de desempenho obtidos pelo modelo do SPARK-NC no OpenMC/Fipy com aqueles obtidos pelo modelo feito no SARAX em [1]. Dentre os diversos resultados mostrados no artigo original que descreve o SPARK-NC, quatro foram escolhidos:

- Evolução temporal do k_{eff} do núcleo entre os tempos $t = 0$ e $t \approx 8.4$ anos, compreendendo a faixa temporal do ciclo total de vida estipulado em [1] para o núcleo à plena potência, que é de 8 anos. (ver Figura 3.1);
- Diferenças de massas dos isótopos ^{235}U , ^{238}U e ^{239}Pu entre o início e o fim do ciclo de vida efetivo do reator à plena potência ($0 \leq t \leq 8$ anos);
- Curva de distribuição axial de potência relativa do núcleo no início do ciclo de vida do reator ($t = 0$);
- Perfis axiais de temperatura em estado estacionário no cerne combustível, revestimento e fluido circundante da vareta mais quente no início do ciclo de vida do reator.

Como os resultados abrangem a faixa temporal correspondente ao ciclo de vida do reator, foi utilizado o módulo de depleção do OpenMC, em que a configuração isotópica do núcleo é atualizada com a transmutação sucessiva dos elementos devido ao processo de fissão e os decaimentos associados conforme a queima do material combustível ao longo desse ciclo. Deve ser mencionado que, devido ao modelo acoplado proposto, o ajuste termohidráulico também é feito durante todo o processo de depleção, acompanhando a modificação do perfil neutrônico do núcleo. Detalhes sobre o processo de depleção e sua implementação no OpenMC podem ser consultados na Seção 3.3.

Em resumo, os dois principais objetivos da dissertação são listados abaixo:

- Modelagem geométrica e de materiais do SPARK-NC na biblioteca OpenMC do *Python* utilizando os dados fornecidos em [1];
- Desenvolvimento de uma metodologia que compreende um módulo termohidráulico capaz de fornecer a distribuição de temperatura do núcleo, além de um algoritmo para a execução de simulações no modelo do SPARK-NC com acoplamento neutrônico e termohidráulico.

Adicionalmente, como objetivo subsidiário, procura-se validar o modelo do SPARK-NC no OpenMC em conjunção com o algoritmo de acoplamento neutrônico/termohidráulico proposto através da comparação de alguns resultados de parâmetros e curvas de desempenho mostrados no artigo original [1] com aqueles obtidos neste trabalho.

1.5 – Estrutura da dissertação

Esta dissertação encontra-se estruturada do seguinte modo:

O Capítulo 1 faz a introdução ao contexto da dissertação em concomitância com uma breve revisão bibliográfica, discorrendo sobre o conceito de SMR e seus principais tipos, os reatores SMR refrigerados à LBE e, mais especificamente, o reator SPARK-NC, que é o reator SMR refrigerado à LBE que é o objeto de estudo deste trabalho. Por fim, os objetivos da dissertação são explicitados;

No Capítulo 2 são dadas as bases teóricas do transporte de nêutrons, apresentando conceitos como o fator de multiplicação efetivo (k_{eff}) e a equação de transporte de nêutrons em sua forma integro-diferencial e integral, relacionando tais conceitos com o método Monte Carlo, que é empregado para cálculos neutrônicos no OpenMC;

O Capítulo 3 é a parte central da dissertação, onde é mostrada a metodologia utilizada para se alcançar os objetivos propostos no Capítulo 1, os métodos analíticos que a embasam, e a sua implementação computacional em códigos como o OpenMC e FiPy;

O Capítulo 4 mostra os resultados (evolução temporal do k_{eff} , diferença de massa de isótopos, perfis de temperatura na vareta mais quente do núcleo, etc.) provindos do modelo computacional do SPARK-NC feito pelo autor na dissertação, comparando-os com os resultados apresentados em [1];

O Capítulo 5 conclui a dissertação, discutindo os resultados alcançados, e propõe possíveis direcionamentos futuros para trabalhos que porventura sucedam o presente texto.

1.6 – Referências do Capítulo

[1] – Hashim, M.; Cao, L.; Zhou, S.; Ma, R.; Shao, Y.; Chen, R. **SPARK-NC: A Lead-Bismuth-Cooled Small Modular Fast Reactor with Natural Circulation and Load Following Capabilities**. *Energies*, Vol. 13, 5410, 21 páginas, 2020.

[2] - OECD-NEA. **Small Modular Reactors: Nuclear Energy Market Potential for Near-term Deployment**, 2016. Disponível em: <https://www.oecd-neo.org/ndd/pubs/2016/7213-smrs.pdf>. Acesso em 28 set. 2023.

- [3] - Liou, Joanne. **What are Small Modular Reactors (SMRs)?**, Disponível em <https://www.iaea.org/newscenter/news/what-are-small-modular-reactors-smrs>. Acesso em 27 set. 2023.
- [4] - World Nuclear Association. **Small Modular Reactors – Key Facts**. Disponível em: <https://www.world-nuclear.org/information-library/nuclear-fuel-cycle/nuclear-power-reactors/small-nuclear-power-reactors.aspx>. Acesso em 31 mar. 2025.
- [5] – AIEA. **Advances in Small Modular Reactor Technology Developments**. Suplemento da *IAEA Advanced Reactors Information System (ARIS)*, 411 páginas, 2022.
- [6] – American Institute of Chemical Engineers (AIChE). **Advances in Very Small Modular Reactors**. Disponível em: <https://www.aiche.org/resources/publications/cep/2022/april/advances-very-small-modular-nuclear-reactors>. Acesso em 31 mar. 2025.
- [7] - Cognet, G.; Bartak, J; Bruna, G. **SMR Safety – Advantages and Challenges**. Anais do *NENE2021*, Bled, Eslovênia, 6-9 de setembro, 2021, 11 páginas.
- [8] - Boado Magan, H.; Halpert, S.; Delmastro, D. F.; Markiewicz, M.; Lopasso, E.; Gimenez, M.; Rauschert, A.; Chocron, M.; Dezzutti, J. C.; Diez, F.; Pirani, H.; & Balbi, C. **CAREM Prototype Construction and Licensing Status**. IAEA, 8 páginas, 2011.
- [9] – Larson, A. **A Closer Look at Two Operational Small Modular Reactor Designs**. Disponível em: <https://www.powermag.com/a-closer-look-at-two-operational-small-modular-reactor-designs>. Acesso em 31 mar. 2025.
- [10] – World Nuclear Association. **Nuclear Power Reactors**. Disponível em: <https://world-nuclear.org/information-library/nuclear-fuel-cycle/nuclear-power-reactors/nuclear-power-reactors>. Acesso em 31 mar. 2025.
- [11] – Reuters. **China launches first commercial onshore small reactor project**. Disponível em: <https://www.reuters.com/world/china/china-launches-first-commercial-onshore-small-reactor-project-2021-07-13/>. Acesso em 2 abr. 2025.
- [12] – World Nuclear News. **Reactor building internal structures completed at chinese SMR**. Disponível em: <https://www.world-nuclear-news.org/Articles/Reactor-building-internal-structures-completed-at>. Acesso em 2 abr. 2025.

- [13] – **Power Reactor Information System (PRIS) – Akademik Lomosonov-1 – Reactor Details.** Disponível em: <https://pris.iaea.org/PRIS/CountryStatistics/ReactorDetails.aspx?current=895>. Acesso em 2 abr. 2025.
- [14] – Rosatom Newsletter. **SMR in the Making.** Disponível em: <https://rosatomnewsletter.com/2020/03/25/smr-in-the-making/>. Acesso em 2 abr. 2025.
- [15] – World Nuclear News. **South Korean SMR design approves by regulator.** Disponível em: <https://www.world-nuclear-news.org/articles/south-korean-smr-design-approved-by-regulator>. Acesso em 2 abr. 2025.
- [16] – Nuclear Regulatory Commission (NRC). **NuScale US460 Standard Design Approval Application Review.** Disponível em <https://www.nrc.gov/reactors/new-reactors/advanced/who-were-working-with/applicant-projects/nuscale-us460.html>. Acesso em 2 abr. 2025.
- [17] – Nuclear Engineering International. **Rolls-Royce SMR begins UK Generic Design Assessment.** Disponível em: <https://www.neimagazine.com/news/rolls-royce-smr-begins-uk-generic-design-assessment-9598877/>. Acesso em 2 abr. 2025.
- [18] – World Nuclear News. **UK regulators begin assessment of Holtec SMR.** Disponível em: <https://www.world-nuclear-news.org/Articles/UK-regulators-begin-assessment-of-Holtec-SMR>. Acesso em 2 abr. 2025.
- [19] – Westinghouse. **Westinghouse Unveils Game-Changing AP300 Small Modular Reactor for Mid-Sized Nuclear Technology.** Disponível em: <https://info.westinghousenuclear.com/news/westinghouse-launches-ap300-smr>. Acessado em 02 abr. 2025.
- [20] – GE Hitachi. **BWRX-300 small modular reactor.** Disponível em: <https://www.gevernova.com/nuclear/carbon-free-power/bwrx-300-small-modular-reactor>. Acesso em 2 abr. 2025.
- [21] – World Nuclear News. **Demonstration HTR-PM connected to grid.** Disponível em: <https://www.world-nuclear-news.org/Articles/Demonstration-HTR-PM-connected-to-grid>. Acesso em 2 abr. 2025.
- [22] – World Nuclear News. **Chinese molten-salt reactor cleared for start up.** Disponível em <https://www.world-nuclear-news.org/Articles/Chinese-molten-salt-reactor-cleared-for-start-up>. Acesso em 2 abr. 2025.

- [23] – World Nuclear News. **IMSR starts second stage of Canadian design review**. Disponível em: <https://www.world-nuclear-news.org/Articles/IMSR-starts-second-stage-of-Canadian-design-review>. Acesso em 2 abr. 2025.
- [24] – World Nuclear News. **Turbine island installation work under way at BREST-OD-300**. Disponível em: <https://www.world-nuclear-news.org/Articles/IMSR-starts-second-stage-of-Canadian-design-review>. Acesso em 2 abr. 2025.
- [25] - Wallenius, J.; Qvist, S.; Mickus, I.; Bortot, S.; Ejenstam, J.; Szakalos, P. **SEALER: A small lead-cooled reactor for power production in the Canadian Arctic**. Anais do *International Conference on Fast Reactors and Related Fuel Cycles: Next Generation Nuclear Systems for Sustainable Development (FR17)*. Yekaterinburgo, Rússia, 26-29 de junho, 2017.
- [26] – Fazio, C. ; Sobolev, V.P.; Aerts, A.; Gavrilov, S.; Lambrinou, K.; Schuurmans, P.; Gessi, A.; Agostini, P.; Ciampichetti, A.; Martinelli, L.; Gosse, S.; Balbaud-Celerier, F.; Courouau, J.L.; Terlain, A.; Li, N.; Glasbrenner, H.; Neuhausen, J.; Heinitz, S.; Zanini, L.; Dai, Y.; Jolkkonen, M.; Kurata, Y.; Obara, T.; Thiolliere, N.; Martin-Munoz, F.J.; Heinzl, A.; Weisenburger, A.; Mueller, G.; Schumacher, G.; Jianu, A.; Pacio, J.; Marocco, L.; Stieglitz, R.; Wetzel, T.; Daubner, M.; Litfin, K.; Vogt, J.B.; Proriol-Serre, I.; Gorse, D.; Eckert, S.; Stefani, F.; Buchenau, D.; Wondrak, T.; Hwang, I.S. **Handbook on Lead-bismuth Eutectic Alloy and Lead Properties, Materials Compatibility, Thermal-hydraulics and Technologies**. Edição 2015. OECD-Nuclear Energy Agency, Paris (França), 2016.
- [27] - Murty, K.L.; Charit.I. **Structural materials for Gen-IV nuclear reactors: Challenges and opportunities**. *Journal of Nuclear Materials*, 383, 7 páginas, 2008.
- [28] – Ahmed, N. M.; Gao, P.; Bello, S. **Natural circulation systems in nuclear reactors: advantages and challenges**. Anais do *IOP Conf. Series: Earth and Environmental Science*, 2020.
- [29] - Usanov, V. I.; Pankratov, D. V.; Popov, É. P.; Markelov, P. I.; Ryabaya, L. D.; Zabrodskaya, S. V. **Long-lived radionuclides of sodium, lead-bismuth, and lead coolants in fast-neutron reactors**. *Atomic Energy*. 87, 5 páginas, 1999.
- [30] - Zrodnikov, A. Chitaykin, V.I; Gromov, B.F. ; Grigoryv, O.G.; Dedoul, A.V.; Toshinsky, G.I; Dragunov, Yu.G.; Stepanov, V.S. **Use of Russian Technology of Ship Reactors With Lead-Bismuth Coolant in Nuclear Power**. Anais do *Advisory Group Meeting on Small Power and Heat Generation Systems on the Basis of Propulsion and Innovative Reactor Technologies*, Obninsk, Rússia, 20–24 de julho, 1998.

- [31] - Dragunov, Yu. G. ; Stepanov, V. S.; Klimov, N. N.; Dedul, A. V.; Bolvanchikov, S. N.; Zrodnikov, A. V.; Tolhinsky, G. I.; Komlev, O. G. **Project of SVBR-75/100 reactor plant with improved safety for nuclear sources of small and medium power.** *Anais do 5th International Conference on Nuclear Option in Countries with Small and Medium Electricity Grids.* Dubrovnik, Croácia, 16-20 de maio, 2004.
- [32] – Hwang, I. S.; Kim, M. H.; Joo, H.J. . **PASCAR - a Small Modular Reactor for PEACER Demonstration.** *Anais do Korean Nuclear Society Spring Meeting.* Gyeongju, Coreia, 29-30 de maio, 2008.
- [33] - Nguyen, T. D. C.; Khandaq, M. F.; Jeong, E.; Choe, J.; Lee, D.; Fynan, D. **MicroURANUS: Core design for long-cycle lead-bismuth-cooled fast reactor for marine applications.** *Energy Research*, 45, páginas, 2021.
- [34] - Sekimoto, H.; Makino, S.; Nakamura, K.; Kamishima, Y. **A long-life small reactor for developing countries, “LSPR”.** *Anais do International Seminar on Status and Prospects for Small and Medium Sized Reactors*, Cairo, Egito, 2001.
- [35] -Zhang, K.; Wang, W.; Duan, C.; Chen, X.; Dua, W.; Luo, X.; Chen, H. **SUMMER: A small modular lead-bismuth-cooled fast reactor for mobile energy supply.** *Progress in Nuclear Energy*, 164, 14 páginas, 2023.
- [36] - Zhou, S; Chen, R.; Shao, Y.; Cao, L.; Bai, B.; Wu, H. **Conceptual core design study of an innovative small transportable lead-bismuth cooled fast reactor (SPARK) for remote power supply.** *International Journal of Energy Research.*, 42, 16 páginas, 2018.
- [37] - Shen, S.; Wang, W.; Chen, H.; Duan, W.; Zhang,. K.; Shi, K.; Chen, Z. **Core design and neutronic analysis of a long-life LBE-cooled fast reactor NCLFR-Oil.** *Progress in Nuclear Energy*, 164, 2023.
- [38] – Zheng, Y.; Du, X.; Xu, Z.; Zhou, S.; Liu, Y.; Wan, C.; Xu, L. **SARAX: A new code for fast reactor analysis part I: Methods.** *Nuclear Engineering and Design*, 340, 10 páginas, 2018.
- [39] – Romano, P. K.; Horelik, N. E.; Herman, B. R.; Nelson, A. G.; Forget, B.; Smith, K. **OpenMC: A State-of-the-Art Monte Carlo Code for Research and Development.** *Annals of Nuclear Energy*, 82, 8 páginas, 2015.

[40] – Guyer, J. E. ; Wheeler, D. ; Warren, J. A. **FiPy: Partial Differential Equations with Python**. *Computing in Science & Engineering*, 10 páginas, 2009.

[41] - Documentação da versão 15.0 do OpenMC - ***docs.openmc.org*** – “*Examples – Functional Expansions*” – Acessado em 05/03/2025.

CAPÍTULO 2

FUNDAMENTOS TEÓRICOS

Este capítulo descreve os fundamentos de transporte de nêutrons e um breve panorama teórico sobre o método de Monte Carlo, que é empregado em códigos como o OpenMC para simular esse transporte de maneira estocástica e extrair grandezas relevantes para a análise das características e do estado de um reator, como é feito neste trabalho para o modelo do SPARK-NC.

2.1 – A equação de transporte de nêutrons

De modo a fornecer uma base teórica sólida, deriva-se inicialmente a equação determinística de transporte de nêutrons; a sua relação com o método de Monte Carlo é mostrada na Seção 2.4. Inicia-se com a grandeza fundamental $N(\vec{r}, \hat{\Omega}, E, t)$, a *densidade de número neutrônico angular*:

$N(\vec{r}, \hat{\Omega}, E, t) d^3r d\Omega dE \equiv$ Número médio de nêutrons no instante t em um volume infinitesimal d^3r em torno do ponto \vec{r} , sendo que cada nêutron em questão possui energia no intervalo $(E, E + dE)$ e move-se na direção da velocidade \vec{v} que atravessa um infinitesimal de área $d\Omega$ determinado pelo vetor de ângulo sólido $\hat{\Omega}$.

O vetor $\hat{\Omega}$ é um vetor unitário cuja origem está no ponto onde o nêutron se localiza, e portanto todas as direções possíveis de $\hat{\Omega}$ para o nêutron definem um lugar geométrico que é uma superfície esférica de raio unitário U_1 com centro no nêutron. Deste modo, pode-se subdividir U_1 em infinitesimais de área $d\Omega$, estando cada um desses infinitesimais em torno do extremo de um dado vetor $\hat{\Omega}$ (é neste sentido que se diz que “ $d\Omega$ é determinado pelo vetor $\hat{\Omega}$ ” na definição de $N(\vec{r}, \hat{\Omega}, E, t)$). Nota-se que, por esta definição, tem-se, dado que U_1 possui raio unitário:

$$\oint_{U_1} d\Omega = 4\pi R^2 = 4\pi.$$

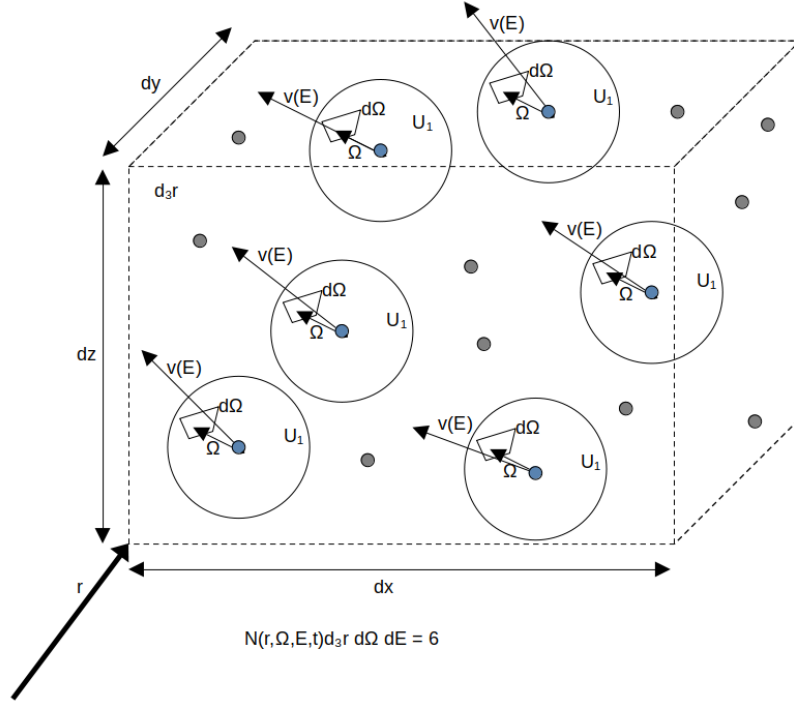
A partir das definições acima, é também possível definir as seguintes grandezas, que são importantes no estudo da neutrônica (aqui, m_n = massa do nêutron):

$\psi(t, \vec{r}, E, \hat{\Omega}) \equiv v(E)N(t, \vec{r}, E, \hat{\Omega})$, com $v(E) = |\vec{v}| = \sqrt{\frac{2E}{m_n}}$, é o *fluxo neutrônico angular*;

$\phi(t, \vec{r}, E) \equiv \oint_{U_1} d\Omega \psi(t, \vec{r}, E, \hat{\Omega})$ é o *fluxo neutrônico escalar* ou somente *fluxo neutrônico*.

A Figura 2.1 mostra, como exemplo, uma situação em que $N(\vec{r}, \hat{\Omega}, E, t) d^3r d\Omega dE = 6$. Nota-se que o vetor \vec{v} de velocidade não precisa ter a mesma direção para cada nêutron, contanto que atravesse a mesma área $d\Omega$ determinada pelo vetor $\hat{\Omega}$ na superfície U_1 com centro em cada nêutron. Entretanto, no limite infinitesimal de $d\Omega$, a diferença entre as direções de $\hat{\Omega}$ e \vec{v} é irrisória e pode-se efetivamente considerar $\vec{v} = v(E)\hat{\Omega}$ para todos esses nêutrons.

Figura 2.1 – Ponto \vec{r} do espaço e instante t em que $N(\vec{r}, \hat{\Omega}, E, t) d^3r d\Omega dE dt = 6$. Os outros nêutrons dentro de d^3r , com direções $\hat{\Omega}'$ e/ou energias E' diferentes de $(\hat{\Omega}, E)$, não são contabilizados.



Define-se também Σ_i , que denota a *seção de choque macroscópica* para o tipo de interação i dos nêutrons com os núcleos atômicos de um dado material: a interação i , neste caso, pode ser uma das diversas interações típicas nêutron-núcleo: espalhamento elástico/inelástico, absorção por captura no núcleo, fissão, etc. Além disso, tem-se a seção de choque total Σ_t , que é a soma de todas as seções de choque referentes a todas as interações possíveis. A seção de choque macroscópica, com unidade de cm^{-1} , é uma espécie de medida de probabilidade de interação dos nêutrons com os núcleos atômicos do material; aqui, não se entrará em detalhes sobre o arcabouço conceitual e teórico das seções de choque, pois existem inúmeras e excelentes referências sobre o assunto, por exemplo [1]. Para a presente discussão, deve apenas ser mencionado que $1/\Sigma_i$ fornece o *caminho livre médio* do nêutron para a interação i naquele material, isto é, a distância média que o nêutron percorre entre duas interações i consecutivas, e, por conseguinte, $1/\Sigma_t$ é o caminho livre médio entre duas interações de qualquer tipo do nêutron no material.

Nota-se que, como as interações nucleares são espacialmente dependentes (pois dependem da composição e densidade atômica do material que preenche um dado ponto no espaço), e podem depender da energia e direção do nêutron incidente, as seções de choque em geral não são constantes, tendo a dependência $\Sigma_i \equiv \Sigma_i(\vec{r}, \hat{\Omega}, E)$. Adicionalmente, no caso de interações que emitem nêutrons, a probabilidade de interação também depende da emissão angular e energética dos nêutrons emergentes, e então geralmente tem-se, para interações $(\hat{\Omega}, E) \rightarrow (\hat{\Omega}', E')$, $\Sigma_i \equiv \Sigma_i(\vec{r}, \hat{\Omega} \rightarrow \hat{\Omega}', E \rightarrow E')$.

A partir do conceito de seção de choque, também define-se outra grandeza importante: Visto que $1/\Sigma_i$ denota a distância média entre interações i , o valor $v(E)/(1/\Sigma_i) = v(E)\Sigma_i$, onde $v(E)$ é a velocidade do nêutron, fornece, naturalmente, a frequência temporal média entre duas interações. Portanto, ao se multiplicar $v(E)\Sigma_i$ pelo número de nêutrons em torno de um dado ponto, tem-se o número médio de interações do tipo i por unidade de tempo em torno daquele ponto (ou o número total médio de interações caso seja $v(E)\Sigma_t$). Assim, o valor

$$\Sigma_i v(E) N(\vec{r}, \hat{\Omega}, E, t) d^3r d\Omega dE \equiv \Sigma_i \psi(\vec{r}, \hat{\Omega}, E, t) d^3r d\Omega dE$$

fornece a *taxa temporal de reação* neutrônica para a interação i no volume d^3r em torno do ponto \vec{r} e no instante t , para nêutrons incidentes com direção e energia em torno de $(\hat{\Omega}, E)$. Caso se deseje a taxa de reação total (isto é, para qualquer interação), $\Sigma_i = \Sigma_t$.

A equação de transporte de nêutrons é tão somente um balanço, que diz que, dentro de um volume fixo ζ , a variação por unidade de tempo, no instante t , do número médio de nêutrons com vetor de direção de movimento em torno de $\hat{\Omega}$ e energia em torno de E é igual à:

1. Quantidade média líquida de nêutrons com direção em torno de $\hat{\Omega}$ e energia em torno de E que entram e saem de ζ , através da fronteira $\partial\zeta$ do volume, por unidade de tempo no instante t ; tal fator entra com um sinal positivo ou negativo no balanço, a depender se, respectivamente, entram mais nêutrons do que saem pela fronteira, ou vice-versa;
2. Quantidade média de nêutrons que é criada com $(\hat{\Omega}, E)$ ou surge pela transformação $(\hat{\Omega}', E') \rightarrow (\hat{\Omega}, E)$, com $\Omega' \neq \Omega$ e/ou $E' \neq E$, dentro de ζ por unidade de tempo no instante t . Tal fator tem o sinal positivo, pois contribui para o incremento de nêutrons $(\hat{\Omega}, E)$ no volume;
3. Quantidade média de nêutrons que é destruída ou sofre a transformação $(\hat{\Omega}, E) \rightarrow (\hat{\Omega}', E')$ dentro de ζ , com $\Omega' \neq \Omega$ e/ou $E' \neq E$, por unidade de tempo no instante t . Tal fator entra com o sinal negativo, pois contribui para o decremento de nêutrons $(\hat{\Omega}, E)$ no volume.

Os próximos parágrafos analisarão cada um desses termos, de modo a compor a equação de transporte. Primeiramente, analisa-se o fator que dá a quantidade média de nêutrons, com direção de movimento em torno de $\hat{\Omega}$ e energia em torno de E , que é destruída ou sofre a transformação $(\hat{\Omega}, E) \rightarrow (\hat{\Omega}', E')$ dentro de ζ , com $\Omega' \neq \Omega$ e/ou $E' \neq E$, por unidade de tempo (item 3). De maneira geral, a seção de choque total Σ_t é a soma das seguintes seções de choque:

- Seção de choque de absorção e captura Σ_a , onde o nêutron é absorvido no núcleo atômico e não participa mais de nenhuma interação; portanto, o nêutron com direção de movimento em torno de $\hat{\Omega}$ e energia em torno de E é, neste caso considerado como “destruído”;
- Seção de choque de espalhamento Σ_s , onde o nêutron com direção de movimento em torno de $\hat{\Omega}$ e energia em torno de E muda para uma direção e/ou energia diferentes $(\hat{\Omega}, E) \rightarrow (\hat{\Omega}', E')$, ou mantêm a mesma direção e energia anteriores, $(\hat{\Omega}, E) \rightarrow (\hat{\Omega}, E)$;
- Seção de choque de fissão Σ_f , onde considera-se que o nêutron incidente é absorvido e produz-se novos nêutrons com um dado espectro de energias e direções de emissão, podendo ser iguais ou diferentes em relação à energia e direção do nêutron incidente $(\hat{\Omega}, E)$.

Deste modo, dado que $\Sigma_t(\vec{r}, \hat{\Omega}, E)\psi(\vec{r}, \hat{\Omega}, E, t) d^3r d\Omega dE$ dá a quantidade total de interações por unidade de tempo $(\hat{\Omega}, E) \rightarrow (\hat{\Omega}', E')$ ponto \vec{r} e no instante t , incluindo interações em que $\Omega' = \Omega$ e $E' = E$, o fator é dado por

$$\left[\int_{\zeta} d^3r \Sigma_t(\vec{r}, \hat{\Omega}, E)\psi(\vec{r}, \hat{\Omega}, E, t) \right] d\Omega dE - n(\Omega, \Omega', E, E'),$$

onde $n(\Omega, \Omega', E, E')$ é a quantidade de interações por unidade de tempo que resultam em $\Omega' = \Omega$ e $E' = E$, que são excluídas pois deseja-se apenas interações onde $\Omega' \neq \Omega$ e/ou $E' \neq E$.

Em seguida, analisa-se o fator que corresponde à quantidade média de nêutrons que é criada com $(\hat{\Omega}, E)$ ou surge pela transformação $(\hat{\Omega}', E') \rightarrow (\hat{\Omega}, E)$ dentro de ζ por unidade de tempo (item 2). A “criação” de nêutrons com $(\hat{\Omega}, E)$, dá-se, em primeiro lugar, pelo processo de fissão, em que um nêutron incidente com $(\hat{\Omega}'', E'')$, com $\hat{\Omega}'' \in U_1$ e $E'' \in (0, \infty)$ resulta na criação de nêutrons $(\hat{\Omega}, E)$. Geralmente, a seção de choque de fissão é angularmente isotrópica, e portanto depende apenas da energia do nêutron incidente, isto é, $\Sigma_f \equiv \Sigma_f(\vec{r}, E'')$. Por outro lado, tal seção de choque deve ser ponderada por uma *função espectral de emissão isotrópica* $\chi(E)/4\pi$ que dá a

probabilidade, dentre todas as energias possíveis com que os nêutrons são emitidos em um processo de fissão, em que o nêutron é emitido apenas com a energia entre E e $E + dE$. O valor de 4π deve-se à isotropia angular, de modo que se tenha probabilidade total igual a 1:

$$\int_0^\infty dE \oint_{U_1} d\Omega \frac{\chi(E)}{4\pi} = \int_0^\infty \chi(E) dE \oint_{U_1} d\Omega \frac{1}{4\pi} = \int_0^\infty dE \chi(E) = 1.$$

Por último, visto que o processo de fissão produz em geral mais de um nêutron, deve-se multiplicar $\chi(E)\Sigma_f(\vec{r}, E'')$ por um valor, designado por ν , que é o número médio de nêutrons emitidos em cada fissão (tal valor situa-se geralmente entre 2 e 3). Tal abordagem é simplificada e considera que todos os nêutrons que podem originar-se de um processo de fissão são produzidos imediatamente (nêutrons prontos). Entretanto, uma abordagem mais realista considera que uma parcela $\beta(E)$ dos nêutrons, sendo essa parcela dependente da energia dos nêutrons emitidos, são produzidos *a posteriori* (nêutrons atrasados), e portanto, teria-se $\nu = \nu(1 - \beta(E))$. Porém, visto que $\beta(E)$ é relativamente pequeno (~ 0.01) [1], não há perda de generalidade na discussão.

Baseado na discussão acima, a taxa de reação para fissão $(\hat{\Omega}'', E'') \rightarrow (\hat{\Omega}, E)$ no volume ζ , ou seja, a quantidade de fissões que criam $(\hat{\Omega}, E)$ no volume é dada por

$$\frac{\chi(E)}{4\pi} \left[\int_{\zeta} d^3r \int_0^\infty dE'' \oint_{U_1} d\Omega'' \nu \Sigma_f(\vec{r}, \hat{\Omega}'', E'') \psi(\vec{r}, \hat{\Omega}'', E'', t) \right] d\Omega dE.$$

Outra forma de “criação” de nêutrons com $(\hat{\Omega}, E)$ é através de fontes de nêutrons; tais fontes são advindas de decaimentos radioativos dentro do volume ζ e são, geralmente, angularmente isotrópicas. Deste modo, define-se a taxa temporal de emissão de uma fonte isotrópica (deve ser notado o fator $1/4\pi$ devido à isotropia angular):

$$\frac{S(\vec{r}, E, t)}{4\pi} d^3r d\Omega dE = \text{Número médio de nêutrons introduzidos por unidade de tempo no instante } t,$$

num volume d^3r em torno do ponto \vec{r} , com energia entre E e $E + dE$ e com o vetor direção de movimento atravessando o infinitesimal de área $d\Omega$, associado com $\hat{\Omega}$, na superfície esférica de raio unitário U_1 com centro em \vec{r} .

Desta forma, a quantidade média de nêutrons criados com $(\hat{\Omega}, E)$ por unidade de tempo, através de uma fonte no volume ζ , é dada por:

$$\left[\int_{\zeta} \frac{S(\vec{r}, E, t)}{4\pi} d^3r \right] d\Omega dE .$$

Outra forma de se introduzir nêutrons $(\hat{\Omega}, E)$ no volume ζ é através de interações de espalhamento entre os nêutrons e o núcleo; neste caso, nêutrons $(\hat{\Omega}', E')$ são espalhados, com possível mudança de direção e/ou energia, tornando-se então nêutrons $(\hat{\Omega}, E)$. Conforme já mencionado, há a possibilidade de espalhamentos $(\hat{\Omega}, E) \rightarrow (\hat{\Omega}, E)$, que não devem ser considerados pois eles não introduzem novos nêutrons $(\hat{\Omega}, E)$ no volume, sendo apenas a continuação temporal da trajetória sem perturbação de um nêutron $(\hat{\Omega}, E)$ já existente, não devendo portanto ser contabilizado no balanço. Desta forma, o termo correspondente à taxa de reação de espalhamento para o fator de criação/introdução de nêutrons $(\hat{\Omega}, E)$ no volume ζ é dado por

$$\left[\int_{\zeta} d^3r \int_0^{\infty} dE'' \oint_{U_1} d\Omega'' \Sigma_s(\vec{r}, \hat{\Omega}'' \rightarrow \hat{\Omega}, E'' \rightarrow E) \psi(\vec{r}, \hat{\Omega}'', E'', t) \right] d\Omega dE - n(\Omega, \Omega', E, E')$$

onde, mais uma vez, o número de espalhamentos $(\hat{\Omega}, E) \rightarrow (\hat{\Omega}, E)$ por unidade de tempo é excluído, pois é uma interação trivial e não contribui para o balanço.

Por fim, analisa-se o fator referente à fuga/entrada de nêutrons $(\hat{\Omega}, E)$ pela fronteira $\partial\zeta$ do volume ζ (item 1). Tal fator é dado por

$$\left[- \oint_{\partial\zeta} N(\vec{r}, \hat{\Omega}, E, t) \vec{v} \cdot d\vec{S} \right] d\Omega dE \equiv \left[- \oint_{\partial\zeta} \psi(\vec{r}, \hat{\Omega}, E, t) \hat{\Omega} \cdot d\vec{S} \right] d\Omega dE,$$

pois $\vec{v} = v(E)\hat{\Omega}$. Aqui, a integral de superfície é tomada sobre a fronteira $\partial\zeta$ do volume e $d\vec{S}$ é o diferencial de área multiplicado pela normal da superfície da fronteira $\partial\zeta$ no ponto \vec{r} , $d\vec{S} \equiv \hat{n}.dS$, que aponta para fora do volume. O sinal negativo existe para que a contribuição seja positiva caso os nêutrons entrem no volume, isto é, se $\hat{\Omega} \cdot d\vec{S} < 0$. Tal integral pode ser transformada em uma integral de volume por meio da seguinte identidade, derivada do Teorema de Gauss:

$$\oint_{\partial\zeta} \varphi d\vec{S} = \int_{\zeta} \nabla \varphi d^3r.$$

Portanto, tem-se, considerando que $\hat{\Omega}$ é independente de \vec{r} :

$$-\oint_{\partial\zeta} \psi(\vec{r}, \hat{\Omega}, E, t) \hat{\Omega} \cdot d\vec{S} = -\hat{\Omega} \cdot \left[\oint_{\partial\zeta} \psi(\vec{r}, \hat{\Omega}, E, t) d\vec{S} \right] = -\int_{\zeta} \hat{\Omega} \cdot \nabla \psi(\vec{r}, \hat{\Omega}, E, t) d^3r.$$

Considerando que a variação, por unidade de tempo, do número médio de nêutrons no volume ζ com vetor de direção de movimento em torno de $\hat{\Omega}$ e energia em torno de E é dada por

$$\left[\int_{\zeta} d^3r \frac{\partial}{\partial t} N(\vec{r}, \hat{\Omega}, E, t) \right] d\Omega dE = \left[\int_{\zeta} d^3r \frac{1}{v(E)} \frac{\partial}{\partial t} \psi(\vec{r}, \hat{\Omega}, E, t) \right] d\Omega dE,$$

tem-se, após somar os fatores do balanço com os sinais adequados, igualar à variação temporal do número de nêutrons e dividir todos esses fatores por $d\Omega dE$:

$$\begin{aligned} \int_{\zeta} d^3r \frac{1}{v(E)} \frac{\partial}{\partial t} \psi(\vec{r}, \hat{\Omega}, E, t) &= -\int_{\zeta} \hat{\Omega} \cdot \nabla \psi(\vec{r}, \hat{\Omega}, E, t) d^3r - \int_{\zeta} d^3r \Sigma_t(\vec{r}, \hat{\Omega}, E) \psi(\vec{r}, \hat{\Omega}, E, t) \\ &+ \int_{\zeta} d^3r \int_0^\infty dE'' \oint_{U_1} d\Omega'' \left[\frac{\chi(E)}{4\pi} \nu \Sigma_f(\vec{r}, \hat{\Omega}'', E'') + \Sigma_s(\vec{r}, \hat{\Omega}'' \rightarrow \hat{\Omega}, E'' \rightarrow E) \right] \psi(\vec{r}, \hat{\Omega}'', E'', t) \\ &+ \int_{\zeta} d^3r \frac{S(\vec{r}, E, t)}{4\pi} \end{aligned}$$

Deve ser notado que, na equação acima, os termos triviais $n(\Omega, \Omega', E, E')$ se anulam, pois um entra com sinal negativo e o outro positivo no balanço. Dado que o volume ζ é arbitrário, a relação também é válida tomando-se em consideração somente os integrandos, e chega-se, portanto, à *equação de transporte de nêutrons*:

$$\begin{aligned} \frac{1}{v(E)} \frac{\partial}{\partial t} \psi(\vec{r}, \hat{\Omega}, E, t) + \hat{\Omega} \cdot \nabla \psi(\vec{r}, \hat{\Omega}, E, t) &= -\Sigma_t(\vec{r}, \hat{\Omega}, E) \psi(\vec{r}, \hat{\Omega}, E, t) + \frac{S(\vec{r}, E, t)}{4\pi} \\ &+ \int_0^\infty dE'' \oint_{U_1} d\Omega'' \left[\frac{\chi(E)}{4\pi} \nu \Sigma_f(\vec{r}, \hat{\Omega}'', E'') + \Sigma_s(\vec{r}, \hat{\Omega}'' \rightarrow \hat{\Omega}, E'' \rightarrow E) \right] \psi(\vec{r}, \hat{\Omega}'', E'', t) \end{aligned}$$

Esta é uma equação integro-diferencial, que fornece o fluxo angular de nêutrons em um ponto arbitrário do espaço de fase $(\vec{r}, \hat{\Omega}, E, t)$, a partir de dados de seção de choque e a distribuição de fontes de nêutrons em tal ponto. A equação de transporte de nêutrons é uma espécie de equação-mestra no estudo da neutrônica; por exemplo, ao se fazer uma série de aproximações em tal equação (com a expansão do fluxo em polinômios de Legendre até ordem 1), deriva-se a celebrada *equação monogrupa de difusão de nêutrons* [1], onipresente no cálculo determinístico em física de reatores:

$$\frac{1}{v} \frac{\partial \phi(\vec{r}, t)}{\partial t} = D \nabla^2 \phi(\vec{r}, t) - \Sigma_a \phi(\vec{r}, t) + S(\vec{r}, t).$$

A equação de transporte de nêutrons é reputadamente de difícil solução em sua forma exata; soluções analíticas existem apenas para as geometrias mais simples e com um alto grau de simetria, ainda assim envolvendo técnicas como expansões em funções especiais ortogonais, equações transcendentais e integração no plano complexo [2]. Deste modo, recorre-se frequentemente à aproximações como a equação de difusão supramencionada, ou à métodos estatísticos como o método de Monte Carlo a ser discutido na Seção 2.4, que não soluciona a equação de transporte diretamente mas oferece um algoritmo estocástico para se obter as integrais \mathcal{T} da forma

$$\mathcal{T} \equiv \int_0^\infty dt \int_\zeta d^3r \int_0^\infty dE \oint_{U_1} d\Omega g(\vec{r}, \hat{\Omega}, E, t) \psi(\vec{r}, \hat{\Omega}, E, t),$$

ou seja, integrais de funções $g(\vec{r}, \hat{\Omega}, E, t)$ ponderadas pelo fluxo $\psi(\vec{r}, \hat{\Omega}, E, t)$, que é a solução da equação de transporte de nêutrons. Uma integral deste tipo é, na terminologia do método Monte Carlo, denominada *tally*, enquanto $g(\vec{r}, \hat{\Omega}, E, t)$ é a função resposta ou *score*.

2.2 – O problema de autovalores para a obtenção do k_{eff}

Dado um sistema multiplicativo, isto é, em que há reações de fissão no domínio do problema, não é garantido que a equação de transporte de nêutrons em estado estacionário e sem fontes externas, isto é, $\frac{\partial}{\partial t} \psi(\vec{r}, \hat{\Omega}, E, t) = 0$, $S(\vec{r}, E, t) = 0$, tenha uma solução não-trivial para uma dada configuração geométrica e de materiais do reator. Porém, pode-se induzir que tal solução exista ao se introduzir artificialmente um termo incógnito $1/k_{eff}$ que “regula” o fator de fissão da equação, condicionando o valor desse fator à existência de uma solução em estado estacionário:

$$\begin{aligned} & \hat{\Omega} \cdot \nabla \psi(\vec{r}, \hat{\Omega}, E) + \Sigma_t(\vec{r}, \hat{\Omega}, E) \psi(\vec{r}, \hat{\Omega}, E) - \int_0^\infty dE'' \oint_{U_1} d\Omega'' \Sigma_s(\vec{r}, \hat{\Omega}'' \rightarrow \hat{\Omega}, E'' \rightarrow E) \psi(\vec{r}, \hat{\Omega}'', E'') \\ &= \frac{1}{k_{eff}} \left[\int_0^\infty dE'' \oint_{U_1} d\Omega'' \frac{\chi(E)}{4\pi} \nu \Sigma_f(\vec{r}, \hat{\Omega}'', E'') \psi(\vec{r}, \hat{\Omega}'', E'') \right]. \end{aligned}$$

Formalmente, ao se definir os operadores

$$\mathcal{L}\psi \equiv \left[\hat{\Omega} \cdot \nabla + \Sigma_t(\vec{r}, \hat{\Omega}, E) \right] \psi(\vec{r}, \hat{\Omega}, E) - \int_0^\infty dE'' \oint_{U_1} d\Omega'' \Sigma_s(\vec{r}, \hat{\Omega}'' \rightarrow \hat{\Omega}, E'' \rightarrow E) \psi(\vec{r}, \hat{\Omega}'', E'')$$

$$\mathcal{F}\psi \equiv \int_0^\infty dE'' \oint_{U_1} d\Omega'' \frac{\chi(E)}{4\pi} \nu \Sigma_f(\vec{r}, \hat{\Omega}'', E'') \psi(\vec{r}, \hat{\Omega}'', E''),$$

pode-se tratar a busca da solução em estado estacionário como um problema de autovalores cuja autofunção é ψ :

$$\{\mathcal{F}^{-1}\mathcal{L}\} \psi(\vec{r}, \hat{\Omega}, E) = \frac{1}{k_{eff}} \psi(\vec{r}, \hat{\Omega}, E).$$

Em análises neutrônicas determinísticas, os valores possíveis de ψ e $1/k_{eff}$ para uma dada configuração podem ser obtidos por métodos numéricos de discretização da equação de transporte, com métodos subsequentes de obtenção de autovalores/autovetores, enquanto no método Monte Carlo uma abordagem distinta é frequentemente utilizada (ver a Seção 2.5). Deve ser mencionado que pode-se ter múltiplos pares (k_{eff}, ψ) para uma dada configuração, mas em geral, se está interessado no par que dá o maior valor de k_{eff} possível, isto é, a solução de *autovalor dominante*.

Apesar de ser um termo artificial, o valor de $1/k_{eff}$ possui um claro significado físico: visto que ele multiplica o fator de fissão, ele determina o quão o número de fissões por unidade de tempo deve aumentar (caso $1/k_{eff} > 1$) ou diminuir (caso $1/k_{eff} < 1$) para que o número de nêutrons no sistema entre em equilíbrio, isto é, para que a quantidade de nêutrons “criados” por unidade de tempo através de fissão (como $S(\vec{r}, E, t) = 0$, não há criação de nêutrons por fontes externas) e aqueles “destruídos” por unidade de tempo através de absorção nos núcleos atômicos ou fuga do domínio do reator sejam as mesmas. Tal condição de equilíbrio é condição *sine qua non* para que haja um estado estacionário, e portanto uma solução da equação de transporte com $\frac{\partial}{\partial t} \psi(\vec{r}, \hat{\Omega}, E, t) = 0$. Caso a configuração do reator já oferece condições para o equilíbrio neutrônico, ($1/k_{eff} = 1$) e não é necessário nenhum termo artificial, pois há de fato uma solução em estado estacionário para a equação de transporte em tal configuração. k_{eff} é denominado, na terminologia da neutrônica, o *fator de multiplicação efetivo* para uma dada configuração do reator e, visto que está relacionado à evolução temporal da população de nêutrons no núcleo do reator para esta configuração, é equivalente à bastante conhecida fórmula dos seis fatores [1].

Observa-se, com base na discussão acima, que o autovalor $1/k_{eff}$ encontrado para uma dada configuração do reator fornece uma medida de *criticalidade* daquela configuração: Se $1/k_{eff} > 1$, ou $k_{eff} < 1$, a configuração necessita de um aumento das fissões para atingir o equilíbrio, indicando que produz uma quantidade de fissões menor que a necessária para que a reação em cadeia seja autossustentável no núcleo; assim, a configuração é *subcrítica*. Por outro lado, se $1/k_{eff} < 1$ ou $k_{eff} > 1$, o número de fissões necessita diminuir para que haja equilíbrio, e

portanto aquela configuração é tal que o número de fissões aumenta com o tempo, sendo uma configuração instável; deste modo, ela é *supercrítica*. Por último, se $1/k_{eff} = 1$ ou $k_{eff} = 1$, a configuração já oferece equilíbrio pois possui uma solução em estado estacionário; é, assim, uma configuração *crítica*. Nota-se também, que o desvio relativo do valor numérico de k_{eff} em relação à unidade quantifica, naturalmente, o desvio da criticalidade; tal grandeza é denominada *reatividade* [1] e dada por $\rho = \frac{k_{eff} - 1}{k_{eff}}$.

2.3 – A forma integral da equação de transporte de nêutrons

A equação de transporte de nêutrons é uma equação integro-diferencial, pois possui operadores de derivada parcial e operadores integrais sobre o fluxo $\psi(\vec{r}, \hat{\Omega}, E, t)$. Entretanto, é possível converter tal equação para uma forma que contém somente operadores integrais. Isto pode ser alcançado ao se multiplicar a equação de transporte pelo fator integrante dependente de um parâmetro s ,

$$I(s, \vec{r}, E, \hat{\Omega}, t) = \exp \left[- \int_0^s \Sigma_t \left(\vec{r} - s' \hat{\Omega}, E, t - \frac{s'}{v(E)} \right) ds' \right],$$

e realizar a integração da equação de transporte sobre uma curva característica do domínio $(t(s), \vec{r}(s))$, eliminando o operador diferencial espaço-temporal $\frac{1}{v(E)} \frac{\partial}{\partial t} + \hat{\Omega} \cdot \nabla$ presente na equação. O desenvolvimento formal desta transformação é demasiadamente extenso, e pode ser encontrado com elevado nível de detalhe na literatura sobre o assunto, por exemplo em [3]. Além disso, a referência [2] mostra uma muito breve, porém excelente, derivação heurística para se obter a forma integral da equação de transporte de nêutrons. Por tais motivos, apresentar-se-á, neste texto, apenas a forma final da equação integral [3]:

$$\begin{aligned} \psi(\vec{r}, \hat{\Omega}, E, t) = & \int_{\Phi_a} d\Phi_a \frac{I(|\vec{r} - \vec{r}_a|, \vec{r}, E_a, \hat{\Omega}_a, t)}{|\vec{r} - \vec{r}_a|^2} Q \left(\vec{r}_a, \hat{\Omega}_a, E_a, t - \frac{|\vec{r} - \vec{r}_a|}{v}, \psi \right) \delta(\hat{\Omega}_a - \hat{\Omega}) \delta(E_a - E) \\ & + \int_{\Phi_a} d\Phi_a \frac{I(|\vec{r} - \vec{r}_a|, \vec{r}, E_a, \hat{\Omega}_a, t)}{|\vec{r} - \vec{r}_a|^2} \frac{S(\vec{r}_a, E_a, t - \frac{|\vec{r} - \vec{r}_a|}{v})}{4\pi} \delta(\hat{\Omega}_a - \hat{\Omega}) \delta(E_a - E) + B(\psi^i, \psi^b), \end{aligned}$$

sendo $B(\psi^i, \psi^b)$ um termo que depende linearmente do fluxo inicial ψ^i em $t = 0$ e do fluxo externo ψ^b que adentra a fronteira do problema para $t > 0$, $\vec{r}_a \equiv \vec{r} - r_a \hat{\Omega}$ com $0 \leq r_a \leq |\vec{r} - \vec{r}_B^a|$, sendo \vec{r}_B^a

o ponto onde \vec{r}_a intercepta a fronteira do problema, $I(|\vec{r} - \vec{r}_a|, \vec{r}, E, \hat{\Omega}_a, t)$ é o fator integrante I definido acima, mas com $s = |\vec{r} - \vec{r}_a|$, $Q(\vec{r}, \hat{\Omega}, E, t, \psi)$ um operador sobre $\psi(\vec{r}, \hat{\Omega}, E, t)$ dado por

$$Q(\vec{r}, \hat{\Omega}, E, t, \psi) = \int_0^\infty dE'' \oint_{U_1} d\Omega'' \Sigma_{sf}(\vec{r}, \hat{\Omega}, \hat{\Omega}'', E, E'') \psi(\vec{r}, \hat{\Omega}'', E'', t),$$

$$\Sigma_{sf}(\vec{r}, \hat{\Omega}, \hat{\Omega}'', E, E'') \equiv \frac{\chi(E)}{4\pi} \nu \Sigma_f(\vec{r}, \hat{\Omega}'', E'') + \Sigma_s(\vec{r}, \hat{\Omega}'' \rightarrow \hat{\Omega}, E'' \rightarrow E),$$

e $\Phi_a = r_a \otimes \hat{\Omega}_a \otimes E_a$ é o *espaço de fase* do problema, com a integral neste espaço de fase dada por

$$\int_{\Phi_a} d\Phi_a \equiv \int_0^{R_B^a} dr_a \int_0^\infty dE_a \oint_{U_1} d\Omega_a,$$

com $R_B^a \equiv |\vec{r} - \vec{r}_B^a|$.

Considera-se agora duas hipóteses plausíveis para um reator nuclear: que não há distribuição de fluxo de nêutrons no domínio em $t = 0$ (início da operação), sendo todo o fluxo de nêutrons introduzido pela fonte $S(\vec{r}, E, t)$ neste instante, e que não há entrada, pela fronteira do núcleo, de um fluxo externo de nêutrons (os nêutrons somente saem pela fronteira do reator, não refletindo de volta). Assim, $\psi^i = \psi^b = 0$, e portanto, usualmente, $B(\psi^i, \psi^b) = 0$ em física de reatores.

A equação integral de transporte é tão complexa quanto a sua forma integro-diferencial; porém, pode ser convertida em uma forma mais palatável, ao se redefinir alguns termos. Primeiramente tem-se

$$\int_{\Phi_a} d\Phi_a \frac{I(|\vec{r} - \vec{r}_a|, \vec{r}, E_a, \hat{\Omega}_a, t)}{|\vec{r} - \vec{r}_a|^2} \frac{S(\vec{r}_a, E_a, t - \frac{|\vec{r} - \vec{r}_a|}{v})}{4\pi} \delta(\hat{\Omega}_a - \hat{\Omega}) \delta(E_a - E) \equiv \hat{S}(\vec{r}, E, t),$$

que é uma fonte em \vec{r} que é a soma ponderada da contribuição da fonte original S em todos os pontos \vec{r}_a e no *tempo retardado* $t - \frac{|\vec{r} - \vec{r}_a|}{v(E_a)}$, onde a ponderação é dada pelo fator

$$\frac{I(|\vec{r} - \vec{r}_a|, \vec{r}, E_a, \hat{\Omega}_a, t)}{|\vec{r} - \vec{r}_a|^2} \delta(\hat{\Omega}_a - \hat{\Omega}) \delta(E_a - E) = \frac{\exp \left[- \int_0^{|\vec{r} - \vec{r}_a|} \Sigma_t \left(\vec{r} - s'' \hat{\Omega}, E, t - \frac{s''}{v(E)} \right) ds'' \right]}{|\vec{r} - \vec{r}_a|^2}.$$

O fator exponencial é denominado *distância óptica*, que é a densidade de probabilidade do nêutron com velocidade $v(E)$ percorrer do ponto \vec{r}_a no tempo retardado $t - \frac{|\vec{r} - \vec{r}_a|}{v(E)}$ até \vec{r} no tempo t através da direção $\hat{\Omega}$ sem sofrer nenhuma interação, enquanto $|\vec{r} - \vec{r}_a|^{-2}$ é um fator de atenuação geométrica devido à dispersão esférica dos nêutrons a partir de uma fonte pontual [3]. Portanto, o termo $\hat{S}(\vec{r}, E, t)$ tem um significado físico claro: é a soma ponderada de todos os nêutrons, das fontes distribuídas no domínio, que têm a possibilidade de alcançar o ponto \vec{r} através da direção $\hat{\Omega}$, sendo essa ponderação dada pela densidade de probabilidade de que tais nêutrons não interajam, e portanto não sejam absorvidos ou mudem suas direções via espalhamento. Em outras palavras, $\hat{S}(\vec{r}, E, t)$ é a contribuição direta (isto é, sem que haja interações), da distribuição de fontes $\frac{S(\vec{r}_a, E_a, t - \frac{|\vec{r} - \vec{r}_a|}{v(E_a)})}{4\pi}$ em todo $\vec{r}_a \equiv \vec{r} - r_a \hat{\Omega}$ com $0 \leq r_a \leq R_B^a$ para o fluxo ψ no ponto \vec{r} , no instante t e na energia E [3].

Da mesma, forma, tem-se para o termo restante, que depende do operador $Q(\vec{r}, \hat{\Omega}, E, t, \psi)$, onde a integral do espaço de fase $\int_{\Phi_a} d\Phi_a$ foi reconvertida para a forma usual

$$\begin{aligned} & \int_0^{R_B^a} dr_a \int_0^\infty dE_a \oint_{U_1} d\Omega_a : \\ & \int_0^{R_B^a} dr_a \oint_{U_1} d\Omega_a \int_0^\infty dE_a \frac{I(|\vec{r} - \vec{r}_a|, \vec{r}_a, E_a, \hat{\Omega}_a, t)}{|\vec{r} - \vec{r}_a|^2} \oint_{U_1} d\Omega'' \int_0^\infty dE'' \Sigma_{sf}(\vec{r}_a, \hat{\Omega}_a, \hat{\Omega}'', E_a, E'') \psi(\vec{r}_a, \hat{\Omega}'', E'', t_r) \\ & \times \delta(\hat{\Omega}_a - \hat{\Omega}) \delta(E_a - E) \\ & = \int_0^{R_B^a} dr_a \int_0^\infty dE'' \oint_{U_1} d\Omega'' \left[\frac{I(|\vec{r} - \vec{r}_a|, \vec{r}_a, E, \hat{\Omega}, t)}{|\vec{r} - \vec{r}_a|^2} \Sigma_{sf}(\vec{r}_a, \hat{\Omega}, \hat{\Omega}'', E, E'') \right] \psi(\vec{r}_a, \hat{\Omega}'', E'', t_r) \\ & = \int_0^{R_B^a} dr_a \int_0^\infty dE'' \oint_{U_1} d\Omega'' K(\vec{r}, \hat{\Omega}, E, t, \vec{r}_a, \hat{\Omega}'', E'') \psi(\vec{r}_a, \hat{\Omega}'', E'', t_r) \\ & = \int_{\Phi_a''} d\Phi_a'' K(\vec{r}, \hat{\Omega}, E, t, \vec{r}_a, \hat{\Omega}'', E'') \psi(\vec{r}_a, \hat{\Omega}'', E'', t_r), \end{aligned}$$

onde $\Phi_a'' = r_a \otimes \hat{\Omega}'' \otimes E''$, e $t_r = t - \frac{|\vec{r} - \vec{r}_a|}{v(E)}$ é o tempo retardado. Aqui,

$K(\vec{r}, \hat{\Omega}, E, t, \vec{r}_a, \hat{\Omega}'', E'')$ é denominado o *kernel de transporte-colisão*. Esta função, com base nas definições dadas acima para a distância óptica atenuada $\frac{I(|\vec{r} - \vec{r}_a|, \vec{r}_a, E, \hat{\Omega}, t)}{|\vec{r} - \vec{r}_a|^2}$ e para a soma das

seções de choque de fissão e espalhamento, Σ_{sf} , também tem um significado intuitivo: É a densidade de probabilidade de que um nêutron que esteja no ponto \vec{r}_a no instante $t - \frac{|\vec{r} - \vec{r}_a|}{v(E)}$ e com energia/direção $(\hat{\Omega}'', E'')$ interaja nesse ponto por espalhamento/fissão de modo que nêutrons com energia E e direção $\hat{\Omega}$ sejam produzidos, e então esses nêutrons produzidos percorram tal direção sem sofrer nenhuma interação até alcançar o ponto \vec{r} no tempo t . Portanto, o termo $\int_{\Phi''_a} d\Phi''_a K(\vec{r}, \hat{\Omega}, E, t, \vec{r}_a, \hat{\Omega}'', E'') \psi(\vec{r}_a, \hat{\Omega}'', E'', t - \frac{|\vec{r} - \vec{r}_a|}{v})$ dá a soma ponderada da distribuição de fluxo em todo $\vec{r}_a \equiv \vec{r} - r_a \hat{\Omega}, E'' \in (0, \infty)$ e $\hat{\Omega}'' \in U_1$, com a ponderação dada pela probabilidade que se tenha uma interação $(\hat{\Omega}'', E'') \rightarrow (\hat{\Omega}, E)$ em \vec{r}_a no instante $t - \frac{|\vec{r} - \vec{r}_a|}{v(E)}$ e os nêutrons produzidos nessa interação não sofram nenhuma interação adicional até alcançar o ponto \vec{r} no instante t de modo a compor o fluxo $\psi(\vec{r}, \hat{\Omega}, E, t)$ [3].

Com a soma dos dois termos já reformatados com as definições acima, a forma integral da equação de nêutrons é colocada na forma canônica de uma equação integral típica, a chamada *equação de Fredholm de segundo tipo* [4]:

$$\psi(\vec{r}, \hat{\Omega}, E, t) = \int_{\Phi''_a} d\Phi''_a K(\vec{r}, \hat{\Omega}, E, t, \vec{r}_a, \hat{\Omega}'', E'') \psi\left(\vec{r}_a, \hat{\Omega}'', E'', t - \frac{|\vec{r} - \vec{r}_a|}{v(E)}\right) + \hat{S}(\vec{r}, E, t).$$

Essa equação integral pode ser solucionada formalmente, ao se substituir, iterativamente, o fluxo ψ dentro da integral pelos termos que estão à direita da igualdade, obtendo a chamada *série de Neumann* [4]:

$$\begin{aligned} \psi(\Phi_0) &= \int_{\Phi_1} d\Phi_1 K(\Phi_0, \Phi_1) \psi(\Phi_1) + \hat{S}(\Phi_0) \implies \\ \psi(\Phi_0) &= \hat{S}(\Phi_0) + \int_{\Phi_1} d\Phi_1 K(\Phi_0, \Phi_1) \hat{S}(\Phi_1) + \int_{\Phi_2} \int_{\Phi_1} d\Phi_2 d\Phi_1 K(\Phi_0, \Phi_1) K(\Phi_1, \Phi_2) \hat{S}(\Phi_2) + \dots \\ \psi(\Phi_0) &= \sum_{n=0}^{\infty} \int_{\Phi_n} d\Phi_n \left[\int_{\Phi_{n-1}} d\Phi_{n-1} \dots \int_{\Phi_1} d\Phi_1 \prod_{k=1}^n K(\Phi_{k-1}, \Phi_k) \right] \hat{S}(\Phi_n). \end{aligned}$$

Aqui, o espaço de fase $\Phi_0 = \vec{r}_0 \otimes \hat{\Omega}_0 \otimes E_0 \otimes t_0$ corresponde aos pontos $(\vec{r}, \hat{\Omega}, E, t)$ onde se deseja medir o fluxo e Φ_i para $i > 0$ é dado por $\vec{r}_i \otimes \hat{\Omega}_i \otimes E_i \otimes t_i^r$, onde $\vec{r}_i = \vec{r}_{i-1} - r_i \hat{\Omega}_{i-1}$, com $0 \leq r_i \leq R_B^i$ e t_i^r é o tempo retardado referente à i interações, isto é,

$t_i^r = t - \sum_{k=1}^i \frac{|\vec{r}_k - \vec{r}_{k-1}|}{v(E_k)} = t - \sum_{k=1}^i \frac{r_k}{v(E_k)}$. Por essas relações, pode-se ter também

$\Phi_i = r_i \otimes \hat{\Omega}_i \otimes E_i$ e portanto $d\Phi_i = dr_i d\hat{\Omega}_i dE_i$. O termo entre colchetes é, de acordo com a definição do *kernel* K dada acima, a probabilidade de um nêutron interagir no ponto do espaço de fase Φ_n , e chegar, após n interações dadas por $(\Phi_n \rightarrow \Phi_{n-1}), (\Phi_{n-1} \rightarrow \Phi_{n-2}) \dots (\Phi_1 \rightarrow \Phi_0)$, até o ponto do espaço de fase Φ_0 , que é onde se quer medir o fluxo. Portanto, em cada termo da série de Neumann, a distribuição de nêutrons por unidade de tempo que são oriundos da fonte $\hat{S}(\Phi_n)$ e que se originam no tempo retardado t_n^r é integrada sobre Φ_n , sendo essa integração ponderada pela probabilidade dos nêutrons realizarem a transição $\Phi_n \rightarrow \Phi_0$ após n interações. Por este motivo, cada um desses termos da série é chamado apropriadamente de *fluxo colidido n vezes* (*n -th collided flux*), $\psi_n(\Phi_0)$ [3], e a série de Neumann é então dada por:

$$\psi(\Phi_0) = \sum_{n=0}^{\infty} \psi_n(\Phi_0),$$

isto é, o fluxo em um ponto $(\vec{r}, \hat{\Omega}, E, t)$ do espaço de fase Φ_0 é a soma de todos os fluxos ψ_n , com $0 \leq n < \infty$, que se originaram de uma fonte e interagiram n vezes de modo que $\Phi_n \rightarrow \Phi_0$ após essas n interações.

Por fim, observa-se que, visto que não há uma distribuição inicial de fluxo em $t = 0$ e não há um fluxo externo adentrando a fronteira, isto é, $B(\psi^i, \psi^b) = 0$, se não há fontes de nêutrons em nenhum local e nenhum instante, isto é, $\hat{S}(\Phi_n) = 0 \forall \Phi_n$, então $\psi_n(\Phi_0) = 0 \forall n$ e $\psi(\Phi_0) = 0$, isto é, não há fluxo de nêutrons no sistema em nenhum local e instante, como seria de se esperar.

2.4 - O método de Monte Carlo para o transporte de nêutrons

A seção anterior mostrou como a equação de transporte de nêutrons pode ser convertida na forma canônica de uma equação integral e, a partir de tal equação, é possível ver que o fluxo ψ em $(\vec{r}, \hat{\Omega}, E, t)$ é, em última instância, a soma de todas as classes ψ_n compreendendo todos os nêutrons que sofreram n interações após serem emitidos pela distribuição de fontes do domínio, chegando até $(\vec{r}, \hat{\Omega}, E, t)$ após essas n interações. Tal perspectiva da obtenção do fluxo neutrônico guarda uma forte similaridade com o método de Monte Carlo para o transporte de nêutrons, que é em essência, a simulação de *histórias* individuais de nêutrons que surgem a partir de uma fonte (seja essa fonte por fissão ou uma fonte externa S) e são transportados através do meio, sofrendo n interações, com a

possível mudança de sua direção e energia $(\hat{\Omega}_i, E_i)$ a cada interação ocorrida em (\vec{r}_i, t_i) , até que esses nêutrons sejam absorvidos ou saiam pela fronteira do domínio sem serem refletidos.

Uma das características fundamentais do método é que tais transições de direção e energia devidas à interação, $(\hat{\Omega}_{i-1}, E_{i-1}) \rightarrow (\hat{\Omega}_i, E_i)$, além do ponto espacial \vec{r}_i onde ocorre cada interação numa dada história de um nêutron são obtidos de maneira estocástica, ao se realizar amostragens sobre funções de densidade de probabilidade $p(x)$ apropriadas que correspondem à física do problema. Neste texto não serão discutidos os detalhes de tal processo de amostragem, visto que a literatura sobre o método é bastante rica acerca do assunto (ver, por exemplo [4]); Aqui, assume-se apenas que qualquer amostragem é *fidedigna* no sentido que o histograma das amostragens de uma variável aleatória x tende à distribuição de densidade de probabilidade $p(x)$ quando o número de amostras tende ao infinito.

Baseado nas discussões dos parágrafos acima, a história de um nêutron é determinada por amostragens apropriadas das seguintes densidades de probabilidade (a terminologia é similar à [4]):

- $p^i(\vec{r}_0, \hat{\Omega}_0, E_0, t_0)$: é a densidade de probabilidade em que o nêutron seja “criado” em qualquer ponto $(\vec{r}_i, \hat{\Omega}_0, E_0, t_i)$, que é quando sua história começa, e chegue até o ponto \vec{r}_0 no instante t_0 , com direção/energia $(\hat{\Omega}_0, E_0)$, sem que sofra nenhuma interação nesse percurso. A “criação” do nêutron dá-se por uma fonte externa ou uma fissão em (r_i, t_i) .
- $p(\vec{r}_k, \hat{\Omega}_k, E_k, t_k | \vec{r}_{k-1}, \hat{\Omega}_{k-1}, E_{k-1}, t_{k-1})$: é a densidade de probabilidade do nêutron interagir no ponto \vec{r}_{k-1} no instante t_{k-1} , onde inicialmente tem direção e energia $(\hat{\Omega}_{k-1}, E_{k-1})$, adquirir a direção e energia $(\hat{\Omega}_k, E_k)$ após a interação, e alcançar, sem nenhuma interação adicional, o ponto \vec{r}_k no instante t_k .
- $p^f(\vec{r}_n, \hat{\Omega}_n, E_n, t_n)$: é a densidade de probabilidade em que o nêutron chegue até o ponto \vec{r}_n no instante t_n , com energia e direção $(E_n, \hat{\Omega}_n)$, e então seja “destruído” nesse ponto por absorção em um núcleo atômico ou por fuga através das fronteiras do domínio do reator, após n interações.

A história ϱ_n de um nêutron que teve n interações antes da sua “destruição” é dada por:

$$\varrho_n : \underbrace{(\vec{r}_i, \hat{\Omega}_0, E_0, t_i) \rightarrow (\vec{r}_0, \hat{\Omega}_0, E_0, t_0)}_{p^i(X_0)} \xrightarrow{p(X_1|X_0)} (\vec{r}_1, \hat{\Omega}_1, E_1, t_1) \xrightarrow{p(X_2|X_1)} \cdots \xrightarrow{p(X_n|X_{n-1})} \underbrace{(\vec{r}_n, \hat{\Omega}_n, E_n, t_n)}_{p^f(X_n)}$$

com a notação abreviada $X_k = (\vec{r}_k, \hat{\Omega}_k, E_k, t_k)$. As densidades de probabilidade de criação, transporte/interação e destruição dependem da geometria do reator e de suas propriedades materiais, via seções de choque macroscópicas. Mais uma vez, os detalhes serão omitidos, podendo ser encontrados em detalhe na literatura especializada (ver, por exemplo, [4]). Uma dada história de um nêutron ϱ_n , com $0 \leq n < \infty$, descreve um processo chamado caminho aleatório (*random walk*), pois a sua criação, transporte, interação e destruição são determinados aleatoriamente por amostragem de funções de densidade de probabilidade.

Será mostrado, para fins ilustrativos, como o método de Monte Carlo é utilizado para estimar uma grandeza obtida através do fluxo integrado em uma dada região do espaço de fase, o chamado *tally*, já apresentado no último parágrafo da Seção 2.1:

$$\mathcal{T} = \int_0^\infty dt \int_\zeta d^3r \int_0^\infty dE \oint_{U_1} d\Omega g(\vec{r}, \hat{\Omega}, E, t) \psi(\vec{r}, \hat{\Omega}, E, t), \text{ ou}$$

$$\mathcal{T} = \int_\zeta d^3r \int_0^\infty dE \oint_{U_1} d\Omega g(\vec{r}, \hat{\Omega}, E) \psi(\vec{r}, \hat{\Omega}, E), \text{ caso o problema esteja em estado estacionário.}$$

Os *tallies*, juntamente com o k_{eff} , estão entre as grandezas mais importantes obtidas em uma simulação neutrônica; diversas variáveis e parâmetros cruciais para a análise de reatores são derivadas da definição geral de um *tally*, ao mudando-se apenas o *score* $g(\vec{r}, \hat{\Omega}, E, t)$; por exemplo, se deseja-se a produção calorífica Q em uma dada região no regime estacionário, que é uma grandeza utilizada na fase de ajuste termohidráulico do SPARK-NC (ver Seção 3.2), tem-se $g(\vec{r}, \hat{\Omega}, E) = h(\vec{r}, E)$, uma função angularmente isotrópica de deposição de calor no material.

De modo a ilustrar uma das maneiras de se estimar tal grandeza pelo método Monte Carlo em um meio não-multiplicativo (isto é, em que não há fissões e portanto $\Sigma_f = 0$), será empregada uma técnica desenvolvida por John Von Neumann e Stanisław Ulam nos anos 1940 [4]. Seja ϱ o conjunto de todas as histórias possíveis de nêutrons numa dada configuração geométrica e de materiais de um reator, isto é todas as realizações possíveis de histórias ϱ_n com n interações antes da “destruição” por absorção ou fuga, sendo $0 \leq n < \infty$. Define-se então uma variável aleatória ξ definida sobre o conjunto ϱ , cujo valor para a história de um nêutron com n interações é dada por:

$$\xi(\varrho_n) = \frac{A(X_0)}{p^i(X_0)} w(X_1, X_0) \cdots w(X_n, X_{n-1}) \frac{q(X_n)}{p^f(X_n)},$$

com

$$w(X_i, X_{i-1}) = \begin{cases} \frac{B(X_i, X_{i-1})}{p(X_i|X_{i-1})}, & \text{se } p(X_i|X_{i-1}) \neq 0 \\ 0, & \text{se } p(X_i|X_{i-1}) = 0 \end{cases},$$

onde, conforme já definido, $X_i = (\vec{r}_i, \hat{\Omega}_i, E_i, t_i)$, ou seja, o ponto do espaço de fase relacionada à i -ésima interação do nêutron, e $A(X)$, $B(X, Y)$ e $q(X)$ são funções quaisquer com as restrições $B(X, Y) \neq 0 \Rightarrow p(X|Y) \neq 0$, $A(X) \neq 0 \Rightarrow p^i(X) \neq 0$ e $q(X) \neq 0 \Rightarrow p^f(X) \neq 0$. Aqui, $p^i(X_0)$, $p^f(X_n)$ e $p(X_i|X_{i-1})$ são, respectivamente, as probabilidades, já definidas, associadas à criação/transporte, destruição e interação/transporte do nêutron em questão. Deve ser notado que, como não há fissões no domínio, as interações antes da “destruição” dos nêutrons são apenas de espalhamento, com a mudança de direção e/ou energia dos mesmos.

A variável ξ é aleatória porque seu valor varia de acordo com a história do nêutron considerada: visto que o nêutron faz um caminho aleatório, os pontos X_i do espaço de fase onde ele é criado, destruído e faz sua i -ésima interação são aleatórios e portanto os valores de densidade de probabilidade $p^i(X_1)$, $p(X_i|X_{i-1})$ e $p^f(X_n)$ entram como parâmetros aleatórios para o cálculo de ξ daquele nêutron. Sendo então ξ uma variável aleatória, ela tem um valor esperado $E[\xi]$ dentro do espaço amostral \mathcal{Q} das histórias possíveis de nêutrons, dado por

$$\begin{aligned} E[\xi] &= \sum_{n=0}^{\infty} \int_{X_0} \cdots \int_{X_n} \xi(X_0 \rightarrow X_1 \rightarrow \cdots \rightarrow X_{n-1} \rightarrow X_n) dP^i(X_0) dP(X_1|X_0) \cdots dP^f(X_n) \\ &= \sum_{n=0}^{\infty} \int_{X_0} \cdots \int_{X_n} dX_0 \cdots dX_n \frac{A(X_0)}{p^i(X_0)} w(X_1, X_0) \cdots w(X_n, X_{n-1}) \frac{q(X_n)}{p^f(X_n)} \times \\ &\quad \times p^i(X_0) p(X_1|X_0) \cdots p(X_n|X_{n-1}) p^f(X_n) \\ &= \sum_{n=0}^{\infty} \int_{X_0} \cdots \int_{X_n} dX_0 \cdots dX_n A(X_0) B(X_1, X_0) \cdots B(X_n, X_{n-1}) q(X_n) \\ &= \sum_{n=0}^{\infty} \int_{X_0} \cdots \int_{X_n} \left[dX_0 \cdots dX_{n-1} A(X_0) \prod_{i=1}^n B(X_{i+1}, X_i) \right] q(X_n) dX_n. \end{aligned}$$

Agora, nota-se que, visto que X_n é o espaço de fase referente à n -ésima interação e Φ_n , definido na Seção 2.3, refere-se ao espaço de fase do nêutron que se situa n interações antes de alcançar o ponto do espaço de fase “atual”, isto é, onde se deseja medir o fluxo, pode-se mapear entre os dois espaços de fase através da relação $X_i = \Phi_{n-i}$. Então, se as funções arbitrárias A , B e q forem substituídas por:

$q(X_n) = g(X_n) = g(\Phi_0)$, $dX_n = d\Phi_0$, $dX_0 \cdots dX_{n-1} = d\Phi_n \cdots d\Phi_1$,
 $B(X_{i+1}, X_i) = K(X_{i+1}, X_i) = K(\Phi_{n-i-1}, \Phi_{n-i}) = K(\Phi_{k-1}, \Phi_k)$, $k = (n - i)$, e
 $A(X_0) = \hat{S}(X_0) = \hat{S}(\Phi_n)$, tem-se:

$$E[\xi] = \int_{\Phi_0} d\Phi_0 g(\Phi_0) \left[\sum_{n=0}^{\infty} \int_{\Phi_n} \cdots \int_{\Phi_1} d\Phi_n \cdots d\Phi_1 \prod_{k=1}^n K(\Phi_{k-1}, \Phi_k) \hat{S}(\Phi_n) \right].$$

O somatório entre colchetes acima corresponde à série de Neumann para a solução da equação integral de transporte de nêutrons dada na Seção 2.3, e portanto:

$$E[\xi] = \int_{\Phi_0} d\Phi_0 g(\Phi_0) \left[\sum_{n=0}^{\infty} \psi_n(\Phi_0) \right] \equiv \int_{\Phi_0} d\Phi_0 g(\Phi_0) \psi(\Phi_0) \equiv \int_{X_n} dX_n g(X_n) \psi(X_n),$$

com $X_n = \Phi_0 = (\vec{r}, \hat{\Omega}, E, t)$. Assim, o valor esperado da variável aleatória ξ , com $q(X)$ substituída pelo *score* $g(X)$ do *tally*, $B(X, Y)$ substituído pelo *kernel* de transporte-colisão $K(X, Y)$ e $A(X)$ substituído por $\hat{S}(Y)$, o termo relacionado à contribuição direta para o fluxo das fontes de nêutrons, como definido na Seção 2.3, dá o *tally* relativo ao *score* g para o fluxo ψ , que é a solução da forma integral da equação determinística de transporte de nêutrons (e também da forma integro-diferencial, pois as mesmas são equivalentes). Assim, além de se mostrar uma técnica para se obter o *tally* relativo a ψ em um caso especial, também demonstra-se, incidentalmente, que o método estocástico de Monte Carlo e o método determinístico para a solução da equação de transporte de nêutrons estão relacionados. Tal relação é natural e esperada, dado que ambas as abordagens descrevem a fenomenologia de transporte de nêutrons na mesma escala espaço-temporal que se situa entre a descrição macroscópica e a descrição microscópica em que efeitos quânticos, como difração, dispersão e superposição de funções de onda, devem ser levados em consideração: tal escala intermediária é denominada *mesoescala*, onde se assume que as grandezas de espaço e tempo são pequenas o suficiente para se adotar uma descrição de *continuum* onde se emprega o cálculo diferencial, mas grandes o suficiente para se desprezar fenômenos quânticos ondulatórios e considerar interações apenas como colisões clássicas de partículas pontuais. O fato de que

$$E[\xi] = \int_{X_n} dX_n g(X_n) \psi(X_n) \equiv \int_0^\infty dt \int_\zeta d^3r \int_0^\infty dE \oint_{U_1} d\Omega g(\vec{r}, \hat{\Omega}, E, t) \psi(\vec{r}, \hat{\Omega}, E, t) \equiv \mathcal{T}$$

faz com que a variável aleatória ξ seja denominada, na terminologia de estatística, um *estimador sem viés* (*unbiased estimator*) do *tally* \mathcal{T} [4].

Deve ser notado que o *kernel* de transporte-colisão $K(X_k, X_{k-1})$ utilizado na abordagem determinística é, conforme a Seção 2.3, a densidade de probabilidade de transição entre os pontos do espaço de fase correspondentes à interação $X_{k-1} \rightarrow X_k$. Então, se é esperado, conforme já mencionado, que as descrições físicas dos fenômenos nas abordagens determinística e estocástica sejam equivalentes, com as densidades de probabilidade de interação/transporte dadas pela geometria e pelas propriedades dos materiais via seções de choque, então $B(X_k, X_{k-1}) = K(X_k, X_{k-1}) = p(X_k|X_{k-1})$, e portanto $w(X_k, X_{k-1}) = 1$, se $p(X_k|X_{k-1}) \neq 0$, e 0 caso contrário. Também, no método Monte Carlo, a unidade dos *tallies* é geralmente normalizada por unidade de partícula-fonte, isto é, o valor da intensidade das fontes S no cálculo do tally é dividido pela quantidade total de nêutrons por unidade de tempo emitidos pelas fontes localizadas em todo o espaço de fase, isto é, dividido por $S_T = \int_X dX S(X)$. Visto que $\hat{S}(X_0)$ dá o número de nêutrons por unidade de tempo criados por fontes externas em qualquer ponto do domínio e que chegam ao ponto do espaço de fase X_0 sem interagir, conforme definido na Seção 2.3, então $\hat{S}(X_0)/S_T$ dá a probabilidade do nêutron ser criado por fontes externas em qualquer ponto e chegar até X_0 sem interagir. Como o sistema é não-multiplicativo e portanto sem fissões, a criação de partículas se dá exclusivamente pelas fontes, e portanto $\hat{S}(X_0)/S_T$ fornece a probabilidade do nêutron ser criado em qualquer ponto e percorrer até o ponto X_0 do espaço de fase sem interagir, isto é, $p^i(X_0)$. Assim, tem-se:

$$A(X_0) = \frac{\hat{S}(X_0)}{S_T} = \frac{\hat{S}(X_0)}{\int_X dX S(X)} = p^i(X_0).$$

Então, ao se substituir $A = p^i$, $w = 1$ e $q = g$ para o valor da variável aleatória ξ para uma dada história de nêutron $\varrho_n \in \boldsymbol{\varrho}$, tem se:

$$\xi(\varrho_n) = \frac{A(X_0)}{p^i(X_0)} w(X_1, X_0) \cdots w(X_n, X_{n-1}) \frac{q(X_n)}{p^f(X_n)} = \frac{g(X_n)}{p^f(X_n)} = g(X_n) \cdot \frac{\Sigma_t(X_n)}{\Sigma_a(X_n)}.$$

A densidade de probabilidade $p^f(X_n)$ do nêutron ser “destruído” em um ponto do espaço de fase X_n é dada pela razão entre a seção de choque de absorção Σ_a e a seção de choque total Σ_t nesse ponto. Aqui considera-se, por simplicidade, que em um ponto do espaço de fase X_n em que \vec{r}_n é pertencente à fronteira do domínio e com $\hat{\Omega}_n \cdot \hat{n} > 0$, onde \hat{n} é o vetor normal da superfície de

fronteira no ponto \vec{r}_n , a seção de choque de “absorção” $\Sigma_a(X_n)$ também abarca a probabilidade de se escapar pela fronteira do domínio.

A partir da técnica mostrada acima, tem-se então um algoritmo relativamente simples pelo método Monte Carlo para se obter o *tally*, com *score* g , referente ao fluxo ψ , que é solução da equação determinística de transporte de nêutrons para a dada configuração de geometria/materiais de um problema não-multiplicativo:

- Simule N histórias de nêutrons até que cada um deles seja absorvido ou saia através da fronteira do domínio do problema, transportando-os através domínio do espaço de fase por meio de amostragens em funções de densidade de probabilidade $p^i(X), p(X_i|X_{i-1})$ e $p^f(X_n)$ que refletem a física do problema;
- Para cada história, registre o ponto do espaço de fase final X_n onde o nêutron foi absorvido ou escapou pela fronteira do domínio, e compute a variável aleatória $\xi = g(X_n)[\Sigma_t(X_n)/\Sigma_a(X_n)]$, onde Σ_a/Σ_t é a probabilidade de absorção (também incluindo a probabilidade de escape caso X_n é um ponto do espaço de fase onde possa ocorrer escape pela fronteira). Armazene, então, o valor de ξ relativo à essa história.
- Após a simulação das N histórias, tem-se uma distribuição de valores da variável aleatória ξ cujo valor esperado é $E[\xi] = \int_{X_n} dX_n g(X_n)\psi(X_n)$, ou seja, o *tally* desejado, normalizado por unidade de partícula-fonte.

As etapas dadas acima descrevem o *algoritmo de Von Neumann-Ulam* [4] aplicado à estimação de *tallies*. Existem vários outros estimadores, como os de colisão (*collision estimator*) e de traço (*track-length estimator*), cada um com suas especificidades, vantagens e desvantagens de aplicação nos mais diferentes problemas. Para mais informações sobre o assunto, [4] é uma excelente referência.

Deve ser observado pela discussão acima que o valor do *tally* é obtido com precisão total, isto é, é o *tally* correspondente ao fluxo ψ que é a solução exata da equação de transporte de nêutrons, apenas se é conhecido com exatidão o valor esperado $E[\xi]$. Isto, porém, é impossível, pois não se conhece *a priori* e com precisão total a distribuição de densidade de probabilidade da variável ξ para uma dada configuração de geometria/materiais onde o problema é definido. Entretanto, existem dois teoremas fundamentais da teoria de probabilidade que fornecem meios de se calcular

uma aproximação para $E[\xi]$ e estimar o quão precisa é essa aproximação em relação ao seu valor exato. O primeiro, denominado **Lei dos Grandes Números**, diz que a média amostral aritmética de ξ feita dentre as N histórias de nêutrons converge para o valor esperado $E[\xi]$ quando $N \rightarrow \infty$.

Apesar da convergência ser garantida pela Lei dos Grandes Números, para uma abordagem eminentemente computacional como o método de Monte Carlo, outra questão de suma importância diz respeito ao quão rápido tal convergência ocorre, isto é, como o número N de histórias de nêutrons simuladas se relaciona com a proximidade da média aritmética de ξ em relação a $E[\xi]$, o que fornece uma medida de compromisso entre demanda computacional e acurácia no cálculo do *tally* \mathcal{T} . Para tal análise, invoca-se o **Teorema do Limite Central**, que, em linhas gerais, diz que, dada uma variável aleatória ξ com valor esperado $E[\xi]$ e desvio padrão σ , a distribuição de densidade de probabilidade do valor da média aritmética desta variável relativa a N amostras, $m(\xi, N)$, tende, quando N tende ao infinito, a uma distribuição Gaussiana com média $\mu = E[\xi]$ e desvio padrão σ/\sqrt{N} :

$$\lim_{N \rightarrow \infty} p(x = m(\xi, N)) = \frac{\sqrt{N}}{\sigma\sqrt{2\pi}} \exp\left(-\frac{N(x - E[\xi])^2}{2\sigma^2}\right).$$

Assim, o teorema não só indica que o desvio padrão da média aritmética em torno de $E[\xi]$ decai aproximadamente como $1/\sqrt{N}$ para um número grande de histórias N , como fornece a informação precisa de quão perto de $E[\xi]$ está o valor da média aritmética dado o valor de N , visto que a distribuição Gaussiana é uma distribuição simétrica em torno de μ com propriedades conhecidas; por exemplo, sabe-se que cerca de 99,7% dos valores da média aritmética $m(\xi, N)$ obtida numa simulação Monte Carlo para o cálculo de um dado *tally* estarão no intervalo $|m(\xi, N) - E[\xi]| < 3\sigma/\sqrt{N}$, para um N muito grande. O Teorema do Limite Central preconiza que tal estimativa é independente do formato da distribuição de densidade de probabilidade associada a ξ (que é fortemente dependente da física do problema e, em geral, bastante complexa).

Apesar do Teorema do Limite Central estabelecer uma taxa de convergência proporcional a $1/\sqrt{N}$ para grandes valores de N , outros fatores de natureza prática devem ser levados em consideração: por exemplo, caso o *tally* deva ser tomado em uma região do espaço de fase muito reduzida, um estimador registrará poucos eventos naquela região (absorção, colisão, etc.) e, por conseguinte, será exigido um esforço computacional muito grande para que se tenha um valor de N significativo para a acurácia da estimação de $E[\xi]$, e portanto, do *tally*. Desta forma, altera-se a simulação original, chamada *análoga* pois a fenomenologia e distribuições de probabilidade associadas às histórias dos nêutrons correspondem à física do problema, para uma simulação *não-*

análoga, onde mecanismos artificiais e não-físicos são introduzidos de modo fazer com que a convergência seja atingida em um tempo de computação menor quando comparado ao da simulação análoga; esses mecanismos são denominados *técnicas de redução de variância*. Um exemplo seria atribuir um fator de importância maior para os nêutrons que tem uma maior chance de serem dirigidos para a região onde se deseja registrar o *tally*, de modo a induzir uma convergência rápida em torno de $E[\xi]$ (no jargão do método Monte Carlo, tal importância é denominada *peso* da partícula). Entretanto, os nêutrons que não são dirigidos para a região do *tally* devem ter seus pesos também modificados, de modo que não haja a introdução de viés estatístico na simulação. Existe uma miríade de referências atinentes a técnicas de redução de variância em métodos Monte Carlo, sendo a referência [4] bastante instrutiva.

2.5 – A obtenção do k_{eff} pelo método Monte Carlo

Nesta seção será dado um breve explanatório sobre um dos algoritmos utilizados para a obtenção do valor de multiplicação efetivo, k_{eff} , definido na Seção 2.2, no método Monte Carlo. Tal algoritmo, denominado *método das gerações sucessivas*, é empregado em códigos como o OpenMC, tendo sido a idéia original do algoritmo publicada em [5]. Começa-se com uma distribuição de fontes inicial S_0 no domínio do problema e, computa-se N_0 histórias de nêutrons gerados a partir desta distribuição, até que todos eles sejam absorvidos ou escapem do domínio. A esses N_0 nêutrons é dado o nome de *geração*. Durante essas histórias, F_0 nêutrons novos são criados por fissão, e um número $N_1 \leq F_0$ desses nêutrons são amostrados, têm seus pontos do espaço de fase iniciais $(\vec{r}, \hat{\Omega}, E)$ armazenados e tornam-se a distribuição de fonte S_1 da próxima geração, cujas histórias geram F_1 nêutrons por fissão, e assim sucessivamente. O autovalor dominante k_{eff} , entre a geração g e a geração $g - 1$ é calculado como

$$k_{eff}^g = \frac{F_g}{F_{g-1}},$$

e o fluxo ψ_g é definido pelas N_g histórias de nêutrons da geração. Isto é feito até atingir o número de gerações (também chamadas de ciclos ou bateladas) especificados na simulação ou até que haja convergência do valor médio do k_{eff} calculado até a geração g dado por

$$\overline{k_{eff}^g} = \frac{1}{g} \sum_{n=0}^g k_{eff}^n, \text{ isto é, } \left| \overline{k_{eff}^g} - \overline{k_{eff}^{g-1}} \right| < \epsilon, \text{ onde } \epsilon \text{ é uma tolerância preestabelecida ou definida}$$

pelo usuário. A amostragem que seleciona N_g nêutrons-fonte da geração g a partir da quantidade F_{g-1} de nêutrons gerados por fissão da geração $g - 1$ é feita para que o número de nêutrons no

domínio não cresça exponencialmente. A distribuição de fontes inicial S_0 é definida pelo usuário, podendo ser feita com base em um *ansatz* ou ser arbitrária; é provado que sempre há convergência para S , que é o valor real de estado estacionário da distribuição de fontes do problema independente de S_0 , sendo a mesma determinante apenas para a rapidez de convergência [5]. Até que haja a convergência $S_0 \rightarrow S$, não é recomendado o cálculo de *tallies*, pois o fluxo não corresponde ao fluxo real ψ para a configuração do problema, e portanto o *tally* pode ser não-físico e/ou impreciso. Assim, necessita-se de um dado número de bateladas *inativas* em que nenhum *tally* é obtido, até que se tenha indicativos que S_g corresponda à distribuição de fontes real em estado estacionário do problema. Em um dos modos de se mensurar tal indicativo, divide-se o domínio em uma malha com n elementos, e, para cada elemento i da malha calcula-se $S_g^{(i)}$, definido como o número de fissões originadas dentro do elemento i dividido pelo número total de fissões no domínio, na batelada g . Então, calcula-se uma métrica comum na teoria da informação, chamada *entropia de Shannon*:

$$H_g = - \sum_{i=1}^n S_g^{(i)} \log_2 S_g^{(i)},$$

verificando a estabilização de tal entropia com $|H_g - H_{g-1}| < \epsilon_H$, que é um indicativo da convergência de S_g para a distribuição real de fontes S no domínio. Informações adicionais sobre o cálculo de k_{eff} no método Monte Carlo podem ser encontradas no artigo original [5].

2.6 – Referências do Capítulo

- [1] – Duderstadt, J. J.; Hamilton, L.J. **Nuclear Reactor Analysis**. Wiley, 1ª Edição, 1976.
- [2] – Tait, J.H. **Introduction to Neutron Transport Theory**. Longman, 1ª Edição, 1964.
- [3] – Prinja, A. K.; Larsen, E. W. **General Principles of Neutron Transport**. Capítulo 5 de: Cacuci, D. **Handbook of Nuclear Engineering**, 111 páginas, Springer, 2010.
- [4] – Spanier, J; Gelbard, E. M. **Monte Carlo Principles and Neutron Transport Problems**. Addison-Wesley, 1ª Edição, 1969.
- [5] – Lieberoth, J. **A Monte Carlo Technique to Solve the Static Eigenvalue Problem of the Boltzmann Transport Equation**. *Nukleonik*, Vol. 11, 7 páginas, 1968.

CAPÍTULO 3

METODOLOGIA E ANÁLISE

Este capítulo tem por objetivo mostrar a metodologia empregada no código para o cálculo de depleção com acoplamento termohidráulico no SPARK-NC, que é o principal objetivo desta dissertação conforme discutido na Seção 1.4. Em linhas gerais, o código pode ser resumido pelas seguintes etapas (os detalhes de cada etapa são fornecidos nas seções indicadas):

Etapla 1. Um modelo do núcleo do SPARK-NC é criado no OpenMC, empregando os dados de geometria e materiais conforme descritos em [1]. Tal modelo é criado como um “objeto” no *Python*, sendo uma instância de uma classe “núcleo” que pode ter configurações diferentes de acordo com os parâmetros passados como argumento durante a criação do mesmo. Um destes parâmetros diz respeito à “granularidade” do modelo do núcleo, isto é, se a geometria do núcleo é subdividida em um maior número de células e múltiplos materiais são criados de modo a preencher as células correspondentes a estas subdivisões. Nesta primeira etapa, este modelo é criado com uma granularidade maior (e é portanto mais complexo), pois deseja-se realizar um ajuste termohidráulico iterativo, chegando-se a valores convergentes de temperatura e densidade do fluido refrigerante (LBE) em diversos pontos do núcleo.

Durante a criação do objeto, o mesmo é carregado com valores iniciais de temperatura e densidade em cada uma de suas células e materiais associados. Visto que as temperaturas das células serão corrigidas durante o ajuste, tais valores de temperatura iniciais podem ser arbitrários (dentro da faixa de temperaturas em que estão especificadas as seções de choque dos materiais) ou aproximados por um *ansatz* sobre o modelo do núcleo. Quanto às densidades, como apenas aquelas correspondentes às células de fluido refrigerante serão atualizadas durante o ajuste termohidráulico através de uma fórmula que relaciona a densidade do LBE com a temperatura (veja Etapa 2), as mesmas podem ser inicializadas de acordo com os valores de temperatura iniciais do LBE por esta fórmula, sendo as demais densidades (material combustível, revestimento, refletor, blindagem, etc.) já determinadas de acordo com o que se espera para uma configuração inicial do núcleo (mais detalhes veja a Seção 3.1).

Etapla 2. Inicia-se uma simulação com o modelo descrito acima com o intuito de calcular o valor do fator de multiplicação efetivo (k_{eff} , ver Subseção 2.2) para este núcleo. O OpenMC possui funções permitindo que parâmetros como temperatura das células e a densidade dos materiais no modelo do núcleo possam ser alterados em tempo de execução (*on-the-fly*): se o cálculo do k_{eff} possui N

ciclos de histórias de partículas, é possível parar a execução em qualquer momento no ciclo $n < N$ e quantas vezes se desejar, de modo a se obter valores de *tallies* e/ou alterar as propriedades do núcleo (temperatura/densidade das células/materiais) e continuar posteriormente a partir daquele ciclo, sendo os ciclos subsequentes sendo calculados com o núcleo já tendo seus parâmetros modificados. O código emprega este conjunto de funções para realizar o ajuste termohidráulico, interrompendo os ciclos de cálculo em intervalos fixos para obter a distribuição axial de densidade de potência de cada vareta combustível naquele momento, realimentando estes dados em um módulo termohidráulico.

O módulo termohidráulico supramencionado tem por objetivo calcular a distribuição de temperatura nas varetas e na região de fluido refrigerante que as envolve, obtendo também a partir desses valores as temperaturas nos elementos não-combustíveis (varetas refletoras, varetas de blindagem, etc.). Ao se obter tais temperaturas, as mesmas são atribuídas às respectivas células, sendo também a densidade do material em cada célula de fluido refrigerante alterada através de uma fórmula determinística que relaciona a mesma com a temperatura - apenas a modificação de densidade do refrigerante é considerada devido ao fato do mesmo ser um fluido e por ter uma maior superfície livre de expansão – portanto com uma variabilidade de densidade maior e com maior capacidade de incorrer em efeitos neutrônicos significativos.

Após os N ciclos, verifica-se a distribuição de temperatura no núcleo convergiu (isto é, se a diferença entre o mapa de temperatura/densidade no ciclo N e do ciclo $N - 1$ é suficiente pequena) e, caso afirmativo, a distribuição de temperatura/densidade atingida para aquele núcleo pode ser considerada a distribuição térmica de estado estacionário induzida pela distribuição neutrônica, também de estado estacionário, associada àquele valor de k_{eff} encontrado. Ao fim desta etapa, são armazenados na memória os valores das temperaturas de cada célula do núcleo, das densidades de cada material associado a uma dada célula de fluido refrigerante e o valor final de k_{eff} obtido (mais detalhes veja a Seção 3.2).

Etapa 3. É criado um novo objeto correspondente ao núcleo do SPARK-NC, porém com um menor refinamento em termos de número de células e materiais. Esta nova versão do núcleo herdará, como condições iniciais, as propriedades termohidráulicas (temperatura e densidade) obtidas do núcleo anterior. Isto é feito através de uma interpolação, dado que a quantidade de células/materiais do núcleo do ajuste termohidráulico é maior do que neste novo núcleo criado.

Com a atribuição de temperaturas/densidades iniciais herdadas do modelo anterior, é feito uma estágio de depleção correspondente a um ano de queima (*burnup*). Após tal cálculo, obtém-se a nova configuração isotópica do núcleo devido aos decaimentos dos produtos de fissão, além do

valor atualizado de k_{eff} após um ano de queima. Ao fim desta etapa, são armazenados na memória um arquivo com a nova configuração isotópica do núcleo, isto é, os elementos presentes em cada célula de combustível onde a queima ocorreu, com suas respectivas abundâncias isotópicas e densidades, calculadas através das equações tendo como parâmetros dados de rendimentos de fissão, e séries de decaimento com suas respectivas razões de ramificação. A escolha por um modelo de núcleo com menor refinamento de células e materiais deve-se ao fato do cálculo de depleção ser intensivo em termos de memória e processamento, pois está acoplado à solução iterativa de um sistema de equações diferenciais matriciais (ver a Subseção 3.3.1), e portanto optou-se por uma menor sobrecarga de recursos computacionais relacionados com um modelo de maior complexidade (mais detalhes veja a Seção 3.3).

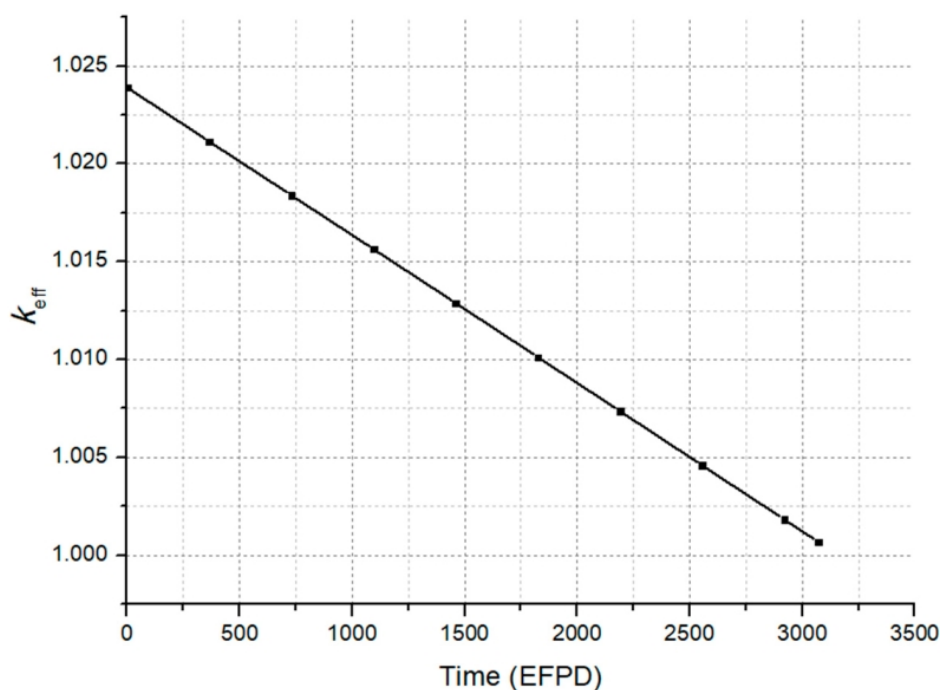
Etapa 4. Um novo objeto núcleo para ajuste termohidráulico é criado (portanto, com um refinamento maior de células/materiais) e a configuração isotópica obtida para cada célula do núcleo da etapa anterior é atribuída à célula correspondente deste novo núcleo através de um mapeamento; deste modo, o núcleo tem a configuração isotópica herdada como “condição inicial” para o ajuste termohidráulico. Um novo estágio de ajuste termohidráulico é executado, e o k_{eff} é calculado enquanto a configuração termohidráulica é ajustada até convergir (veja Etapa 2). Após o ajuste, as temperaturas de cada célula e as densidades dos materiais de cada célula correspondente ao fluido refrigerante são armazenadas, assim como o valor do k_{eff} encontrado.

De posse de tais dados termohidráulicos, um novo objeto de núcleo para depleção mais simples que o núcleo de ajuste termohidráulico é criado, e as temperaturas, densidades e configuração isotópica obtida na Etapa 3 são carregadas neste novo objeto, realizando mais um cálculo de depleção correspondente a um ano de *burnup* (o que, na prática, representa a depleção que ocorre entre o primeiro e o segundo ano de operação do SPARK-NC, visto que esta depleção é feita com o núcleo em configuração isotópica obtida ao fim de um ano de depleção na etapa anterior). Após a depleção, uma nova configuração isotópica é obtida (correspondente a dois anos de *burnup*) e tais dados são carregados em um novo objeto de núcleo para ajuste termohidráulico, com as simulações de ajuste termohidráulico/depleção ocorrendo de maneira sucessiva até que se complete os 8.4 anos de operação do SPARK-NC como previsto em [1] (para isto, nota-se que a última etapa de depleção deve ser de 0.4 anos e não de 1 ano como as anteriores).

Ao final da simulação, deseja-se obter dois grupos de dados para a comparação com os resultados encontrados em [1]:

- o k_{eff} do núcleo em cada ano de depleção (do “ano 0”, ou seja, o início da operação, até o “ano 8.4”, correspondente ao ponto final da abscissa do gráfico da Figura 3.1, quando o ciclo de vida do combustível no reator efetivamente termina com o k_{eff} próximo da criticalidade ($k_{eff} = 1$).
- As diferenças das massas dos isótopos ^{235}U , ^{238}U e ^{239}Pu após o fim do ciclo de vida do núcleo, isto é, entre os tempos $t = 0$ e $t = 8$ anos (dado que o tempo de vida do combustível considerado é de 8 *EFPY* – *Effective Full Power Years* [1]), que conforme [1], são de, respectivamente, 107.5 kg, 63.1 kg e -51.4 kg (o valor negativo indica que houve produção de Plutônio-239).
-

Figura 3.1 – Evolução do k_{eff} do núcleo ao longo do tempo devido à depleção. Aqui, EPFD significa dias efetivos à plena potência (*Effective Full Power Days*). A distância entre dois pontos consecutivos corresponde a 1 ano, exceto entre o último e o penúltimo ponto, onde corresponde a cerca de 0.4 ano. Retirado de [1].



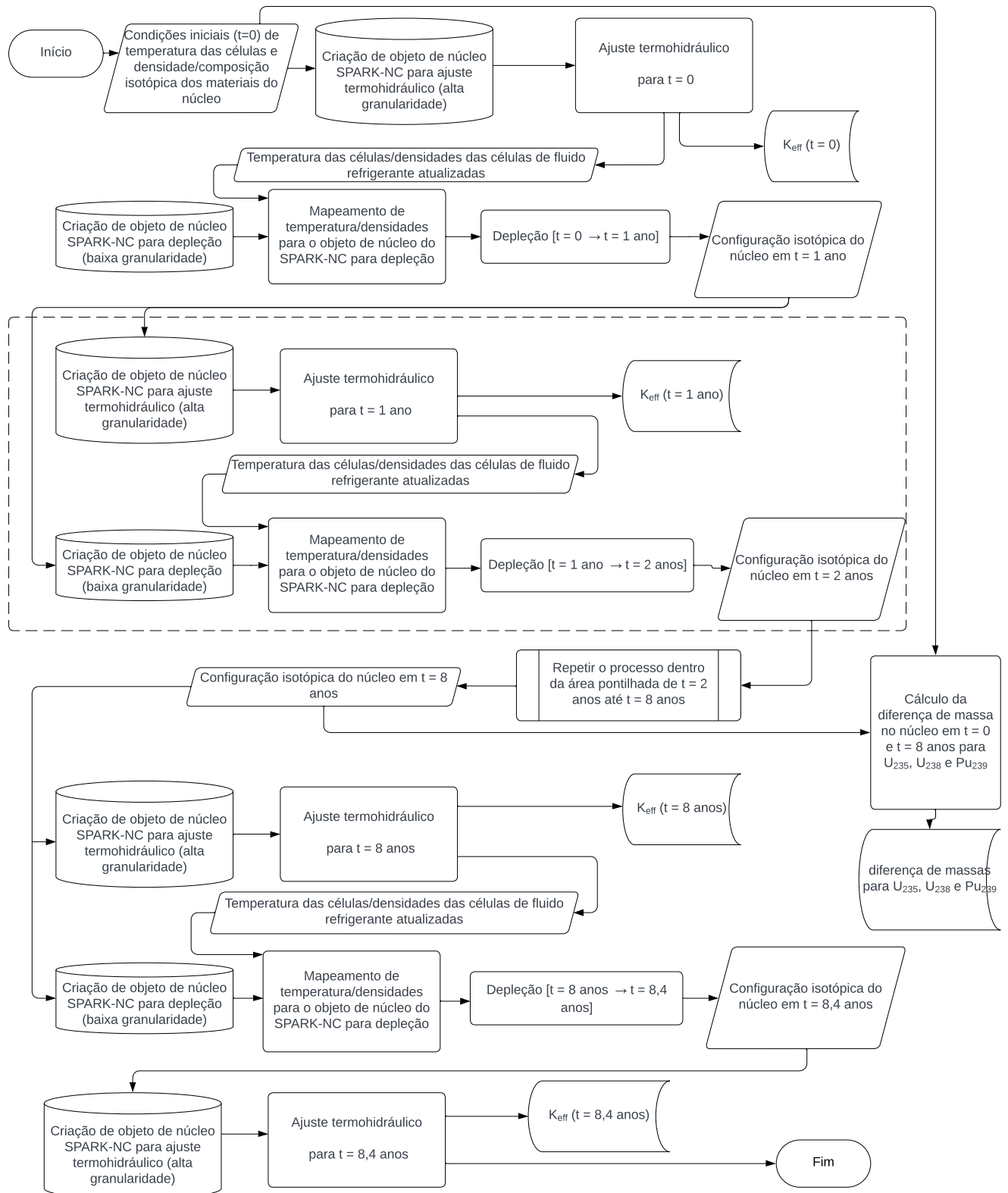
Os valores de k_{eff} utilizados para comparação com o gráfico da Figura 3.1 em um dado ano N são aqueles obtidos ao final de cada estágio de ajuste termohidráulico imediatamente posterior ao estágio de depleção correspondente à operação entre o ano $N - 1$ e o ano N , pois são os valores considerados mais precisos: é o k_{eff} encontrado para o núcleo com a configuração isotópica do ano N (pois foi carregado com tal configuração advinda do estágio de depleção $[N - 1, N]$) e com o núcleo tendo a distribuição de temperatura/densidades consideradas de estado estacionário para esta

configuração isotópica. Nota-se que o k_{eff} empregado para a comparação com o ano 0 do gráfico é aquele obtido no estágio de ajuste termohidráulico inicial (Etapa 1 descrita acima), pois não existe configuração isotópica advinda de cálculo de depleção anterior, apenas a configuração inicial do núcleo.

Os valores de diferenças de massa, por sua vez, são obtidos através dos dados de configuração isotópica do combustível em $t = 0$ e $t = 8$ anos, onde o OpenMC torna possível extrair, para uma dada configuração, as massas totais no núcleo de um dado isótopo.

Um fluxograma da metodologia é mostrado na Figura 3.2. Deve ser notado que, para um dado ano N na simulação, ambos os estágios (ajuste termohidráulico e depleção) fazem o modelo do núcleo “avançar no tempo” de maneiras distintas: enquanto o estágio de depleção atualiza a configuração isotópica do núcleo do ano $N - 1$ para o ano N considerando como estática a sua distribuição de temperatura/densidade que foi calculada no estágio anterior (ano $N - 1$) e que lhe é dada como condição inicial, o ciclo de ajuste termohidráulico atualiza a distribuição de temperatura/densidade para aquela configuração isotópica obtida, levando ambas as configurações (isotópica e termohidráulica) para o mesmo ponto temporal (ano N).

Figura 3.2 – Fluxograma do processo geral de depleção/ajuste termohidráulico, com a geração do k_{eff} para t entre 0 e 8.4 anos e comparação das diferenças de massa de ^{235}U , ^{238}U e ^{239}Pu entre $t = 0$ e $t = 8$ anos. Observa-se que o valor de k_{eff} considerado é aquele gerado ao final de cada estágio de ajuste termohidráulico.



3.1 – A modelagem do núcleo do SPARK-NC no OpenMC

Esta seção tem por objetivo descrever o processo de modelagem do núcleo do SPARK-NC no OpenMC. Um panorama da biblioteca OpenMC para o *Python*, além da descrição de suas principais funcionalidades, são dados na Seção 3.2. Deste modo, a discussão na presente seção irá ater-se tão somente aos aspectos mais gerais da modelagem e implementação, sem adentrar de maneira minuciosa em detalhes sobre o código.

3.1.1 – A arquitetura do núcleo do SPARK-NC

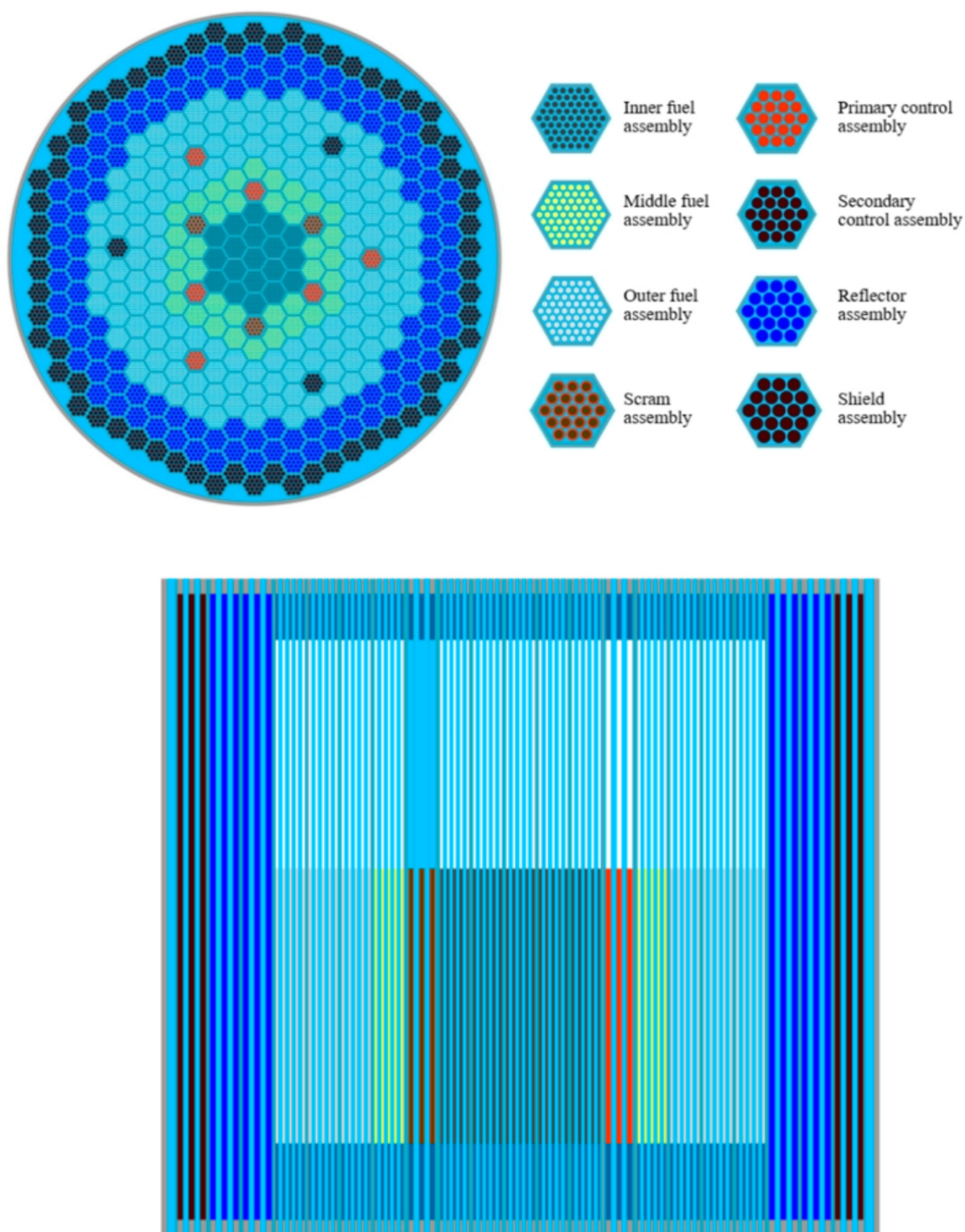
O SPARK-NC foi brevemente descrito na Seção 1.3, sem que se aprofundasse em detalhes de sua arquitetura, como a geometria, a composição e características de especificação. Esta seção apresenta estas informações, que são empregadas na construção do modelo do núcleo utilizando as funções básicas da biblioteca do OpenMC.

Como é típico de um reator SMR, o núcleo do SPARK-NC é um núcleo compacto, com o diâmetro do vaso do reator contendo apenas 232 cm de diâmetro (com 2 cm de espessura das paredes do vaso) e de potência relativamente reduzida (30 MW térmicos). O seu arranjo é composto por 8 tipos de elementos de varetas, com funcionalidades e características distintas [1]:

- Três tipos de elementos combustíveis (localizados nos “anéis” interno, médio e externo do núcleo, contendo um número de 19, 36 e 132 elementos de cada tipo, respectivamente);
- Dois tipos de elementos de controle (primário, com 6 elementos e secundário, com 3 elementos);
- 3 elementos de SCRAM,;
- 114 elementos refletores;
- 66 elementos de blindagem.

Os elementos possuem seção de corte transversal hexagonal, conforme pode ser notado na Figura 3.3, que mostra a configuração radial e axial do núcleo. Observa-se também que as varetas dos elementos combustíveis possuem um diâmetro menor que as dos outros tipos de elementos (as características geométricas dos elementos serão fornecidas no decorrer desta seção).

Figura 3.3– Configuração radial e axial do núcleo do SPARK-NC. Retirado de [1].



Os elementos combustíveis estão localizados, conforme já mencionado e observado na Figura 3.3, em três regiões anulares concêntricas ao centro geométrico do núcleo. Cada um dos três tipos de elementos referentes aos anéis interno, médio e externo possuem características distintas geométricas e de composição do combustível, de acordo com a Tabela 3.1. Nota-se que tais elementos possuem uma relação *pitch*/diâmetro das varetas relativamente alto, de modo a facilitar a

circulação natural do fluido refrigerante (Chumbo-Bismuto Eutético) presente no projeto do SPARN-NC [1], que circula no núcleo com uma temperatura de entrada de 330 °C (acima da temperatura de fusão do LBE que é de cerca de 123.5 °C) e uma velocidade de 0.31 m/s em operação nominal [1].

Tabela 3.1 – Especificações dos elementos combustíveis do SPARK-NC [1].

Item	Especificação
Percentual de enriquecimento do Dióxido de Urânio (UO_2) - elementos dos anéis interno / médio / externo	17.00 / 17.00 / 19.75
Material de revestimento das varetas e elementos	Aço T91
Material de preenchimento do <i>gap</i>	Hélio
Distância plano-a-plano entre duas paredes externas paralelas do elemento hexagonal (cm)	10.6
Espessura do revestimento do elemento (cm)	0.3
Número de varetas por elemento	61
Diâmetro das varetas (interno / médio / externo, em cm)	0.96 / 1.0 / 1.0
Relação <i>pitch</i> /diâmetro (interno / médio / externo)	1.28 / 1.24 / 1.24
Espessura dos revestimento das varetas (cm)	0.04
Espessura do <i>gap</i> (cm)	0.015
Altura ativa da vareta (cm)	90
Altura do <i>plenum</i> (cm)	75
Altura do refletor superior (cm)	15
Altura do refletor inferior (cm)	25
Altura das tampas inferior e superior da vareta (cm)	5

O combustível utilizado nas varetas é o Dióxido de Urânio (UO_2) com enriquecimento entre 17% e 19.75% (o maior enriquecimento é alocado para as varetas do anel mais externo, de modo a permitir uma distribuição mais homogênea de potência no núcleo) e o material escolhido para o revestimento das varetas é o aço T91, devido às propriedades mecânicas adequadas para o projeto

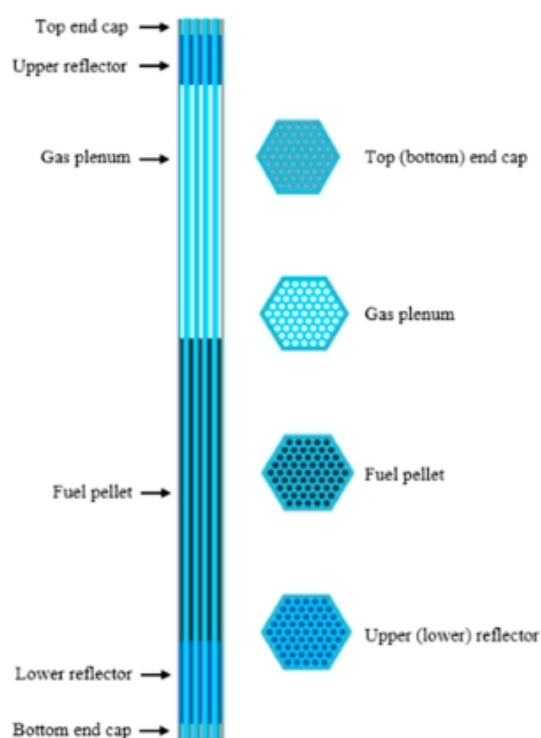
do SPARK-NC, como a resistência à corrosão pelo LBE [1]. A composição do aço T91 utilizada no modelo do núcleo não foi explicitada pelos autores do artigo original em [1]. A literatura mostra várias composições distintas, porém com um alto grau de similaridade, e o autor da dissertação emprega em seu modelo do núcleo a composição (em porcentagem de fração atômica) mostrada na Tabela 3.2, que por sua vez provém da referência [2], que discorre acerca da interação físico-química do LBE com diversas ligas de metal (incluindo o T91).

Tabela 3.2 – Composição típica do aço T91, conforme [2].

Elemento	Composição (%)	Elemento	Composição (%)
Ferro	88.65	Nióbio	0.04
Manganês	0.47	Carbono	0.46
Cromo	9.06	Vanádio	0.23
Molibdênio	0.55	Níquel	0.11
Silício	0.43		

Quanto à geometria, as varetas combustíveis possuem as seguintes regiões (Ver Figura 3.4): o revestimento lateral, as tampas inferior e superior (*top/bottom end cap*, composto por aço T91), combustível (*fuel pellet*, UO_2 enriquecido), *gap* revestimento-combustível e *plenum* para acúmulo de produtos de fissão (ambos preenchidos com Hélio) e refletores inferior e superior (*upper/lower reflector*). Para estes dois últimos, os autores em [1] não fornecem o material utilizado, tendo fornecido apenas a composição das varetas dos elementos puramente refletores (*reflector assemblies*, onde mencionam que os mesmos são compostos por varetas maciças de aço T91). Foi assumido que o material dos refletores é único, e portanto o material dos refletores inferior e superior das varetas também é composto de aço T91. Tal hipótese é plausível, visto que o T91 foi considerado como refletor para as varetas e para os elementos refletores na versão inicial do SPARK [3][4], com valores da evolução de k_{eff} no processo de depleção bastante similares aos do SPARK-NC (comparar a Figura 3.1 e a Figura 3 em [3]).

Figura 3.4 – Corte longitudinal e cortes transversais mostrando a geometria dos elementos combustíveis. Retirado de [1].



Conforme já mencionado, o SPARK-NC apresenta dois sistemas independentes de controle (primário e secundário), além de um sistema de desligamento de emergência (SCRAM). Enquanto o sistema primário (6 elementos) é responsável pelo controle de reatividade e por homogeneizar a distribuição radial de potência, o sistema secundário é responsável pelo ajuste fino da dinâmica de potência no núcleo. Os três sistemas utilizam o B_4C (Carbeto de Boro) como elemento absorvedor, com diferentes enriquecimentos relativos ao isótopo ^{10}B . A montagem dos três elementos é similar, contendo revestimento e tampas de T91, refletores inferiores e superiores (também de aço T91), região absorvedora e *plenum*. Enquanto nos dois sistemas de controle o *plenum* é idêntico ao dos elementos combustíveis, sendo preenchidos por gás hélio, os elementos de SCRAM têm seus *plena* preenchidos por LBE.

Ao contrário da região absorvedora das varetas de controle que é composta apenas por pastilhas de B_4C , as varetas nos elementos de SCRAM são compostas por lastros de Tungstênio para conter a alta fluatibilidade do LBE e permitir uma rápida descida em emergência, com esses lastros cilíndricos sendo apenas revestidos por uma camada de B_4C . As características geométricas e de composição dos elementos de controle/SCRAM são mostrados na Tabela 3.3 e um corte transversal de tais elementos pode ser visto na Figura 3.6. Deve ser observado que, devido à necessidade de se mover livremente os elementos de controle/SCRAM para inserção/retirada do núcleo, esses elementos são compostos por uma casca prismática hexagonal externa fixa no núcleo

(que possui as mesmas dimensões do revestimento dos outros tipos de elemento) e outro prisma hexagonal interno, contendo a estrutura com as varetas absorvedoras e que é capaz de deslizar verticalmente no interior dessa casca prismática. Entre os prismas externo e interno existe um espaçamento, que fica preenchido com LBE (ver Figura 3.5).

Tabela 3.3 – Especificações dos elementos de controle/SCRAM do SPARK-NC [1].

Item	Especificação
Composição do absorvedor (interno /médio / externo), em porcentagem de enriquecimento do isótopo ^{10}B no B_4C	40.00 / 19.99 / 19.99
Material de revestimento das varetas e elementos	Aço T91
Material de preenchimento do <i>gap</i>	Hélio
Distância plano-a-plano entre duas paredes paralelas externas da estrutura fixa hexagonal (cm)	10.6
Espessura do revestimento da estrutura fixa do elemento (cm)	0.3
Espessura do revestimento da parte deslizante do elemento (cm)	0.2
Espessura do espaçamento entre parte fixa e parte deslizante (cm)	0.2
Número de varetas por elemento	19
Raio das varetas (interno /médio / externo, em cm)	0.9 / 0.9 / 0.6
<i>Pitch</i> do elemento	2.01
Espessura dos revestimento das varetas (cm)	0.04
Espessura do <i>gap</i> (cm)	0,02
Altura ativa da vareta (cm)	90
Altura do <i>plenum</i> (cm)	75
Altura do refletor superior (cm)	15
Altura do refletor inferior (cm)	25
Altura das tampas inferior e superior da vareta (cm)	5

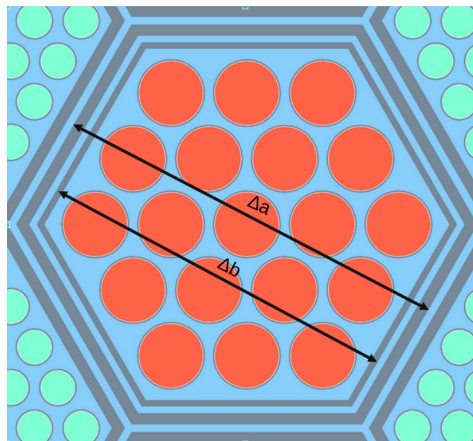
Os elementos refletores e de blindagem do núcleo são compostos por 19 varetas cada, apresentando uma montagem relativamente mais simples: as varetas de blindagem são compostas

por cernes de B_4C em composição isotópica natural (19% de ^{10}B) revestidos por aço T91, enquanto os elementos refletores são varetas maciças de aço T91 [1].

Uma observação acurada da Figura 3.3 mostra que, apesar de ter o mesmo número de varetas dos elementos de controle/SCRAM, os diâmetros e *pitch* das varetas desses elementos são menores que os das varetas refletoras e de blindagem. Entretanto, os valores de tais grandezas para esses últimos elementos não são mostrados em [1]. Como tais grandezas podem ter um impacto não-desprezível sobre a neutrônica, devido à quantidade total de varetas dos dois tipos, empregou-se um fator de conversão: Tomou-se a razão entre as distâncias Δa e Δb , ou seja, entre a distância plano-a-plano da parede externa da casca prismática hexagonal pelo qual o elemento desliza e a distância plano-a-plano do elemento em si (ver Figura 3.5). Adotou-se tal fator visto que, apesar ter o mesmo número de varetas, tal espaçamento é inexistente nos elementos de blindagem e refletores. O fator então foi multiplicado pelo valor de *pitch* (2.01 cm) e pelo maior diâmetro das varetas dos elementos de controle/SCRAM (1.8 cm) para se obter os respectivos valores de *pitch* e diâmetro para as varetas dos elementos de blindagem e refletores. De acordo com a Tabela 3.2, o valor do fator é dado por:

$$\delta = \frac{\Delta a}{\Delta b} = \frac{10.6}{10.6 - 2(0.3) - 2(0.4)} \simeq 1.1.$$

Figura 3.5 – Relação entre a distância plano-a-plano da superfície externa do elemento hexagonal de controle/SCRAM (Δb), e da distância plano-a-plano da superfície externa da casca prismática hexagonal pelo qual o elemento desliza (Δa).

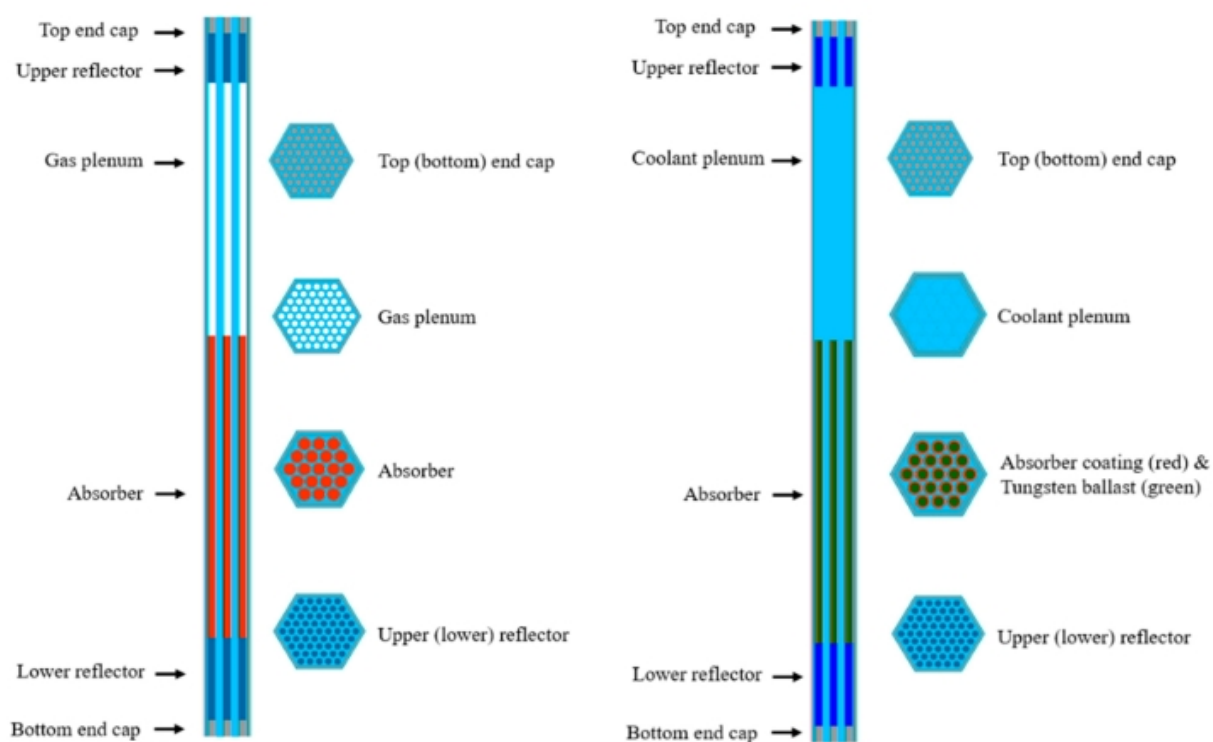


Algumas especificações importantes sobre o núcleo do SPARK-NC, fornecidas em [1] e que serão empregadas nesta dissertação são apresentadas na Tabela 3.4.

Tabela 3.4– Especificações importantes para o núcleo do SPARK-NC, empregados nesta dissertação [1].

Item	Especificação
Temperatura de entrada do LBE no núcleo (°C)	330
Inventário de metal pesado no início da vida útil do reator (kg)	5745
Velocidade média do LBE no núcleo em estado estacionário (m/s)	0.31
Potência do Reator (MWt)	30
Ciclo de vida do combustível (anos efetivos em potência máxima)	8
Fluxo de massa do fluido refrigerante no núcleo (kg/s)	1746
<i>Pitch</i> dos elementos no núcleo (cm)	10.8
Diâmetro do vaso do reator (interno/externo, em cm)	232/236
Altura do núcleo (cm)	215
Número de elementos combustíveis (interno /médio / externo)	19/36/132
Números de elementos de controle (secundário / primário / SCRAM)	6/3/3
Número de elementos refletores/blindagem	114/66

Figura 3.6 – Cortes transversais e longitudinais dos elementos de controle (esq.) e SCRAM (dir.) Retirado de [1].



3.1.2 – A modelagem da arquitetura do núcleo do SPARK-NC no OpenMC

Esta seção mostra como a arquitetura do SPARK-NC detalhada em 3.1.1 foi implementada, através de suas especificações, no *Python* com a biblioteca OpenMC. De modo geral, a estrutura de código para a criação da arquitetura envolveu a definição de algumas classes *Python* pelo autor:

- Classe **CoreAssemblies** – É a classe responsável pela criação dos objetos relativos aos elementos (combustível, controle, refletores, blindagem, etc.), que serão replicados em um reticulado hexagonal para a composição do núcleo. Tal classe, ao ser instanciada, recebe como entrada o tipo de elemento a ser criado, além de parâmetros geométricos (*pitch*/diâmetro das varetas, espessura de revestimentos, altura das partes da vareta, etc.) e de composição material do elemento. A classe também recebe, para cada um dos componentes do elemento, informações sobre seu particionamento, isto é, em quantas células aquele componente deve ser subdividido – isto está relacionado ao grau de refinamento do núcleo explanado na Etapa 1 no início do Capítulo 3.
- Classe **CoreMaterials** – Esta classe cria o inventário de materiais que irá preencher cada uma das células de cada um dos componentes dos elementos do núcleo. Ela recebe como dados de entrada as densidades e composições de cada material necessário para o preenchimento das células. Durante a criação de um dado objeto correspondente ao núcleo do SPARK, é necessária a criação de um único objeto **CoreMaterials**, pois ele fornecerá informações sobre os materiais de qualquer um dos componentes de qualquer tipo de elemento no núcleo.
- Classe **ReactorCore** – Esta é a classe que irá instanciar os objetos da classe **CoreAssemblies** relativos a todos os tipos de elementos, atribuir a esses objetos os materiais associados do objeto **CoreMaterials** e replicar esses objetos em um reticulado hexagonal de acordo com o arranjo geométrico do núcleo do SPARK-NC. Ela também é responsável por criar as células correspondentes à parede do vaso do reator e o fluido refrigerante que circunda o arranjo de elementos hexagonais. Portanto, essa classe encapsula a criação de um objeto correspondente a um núcleo completo do SPARK-NC.

Cada uma das classes será explicada com maior propriedade nas subseções seguintes, incluindo a discussão sobre algumas funções subsidiárias utilizadas por essas classes.

Adicionalmente, o código para o modelo neutrônico/termohidráulico acoplado produzido para a dissertação possui quatro módulos criados pelo autor (um módulo no *Python* é um arquivo contendo estruturas de dados ou definições de funções/classes que pode ser “carregado” em outro arquivo, como uma biblioteca). Tal estratégia foi adotada de modo a reunir em arquivos separados funções, classes e dados que possuíssem características similares, de modo a melhorar a legibilidade do código. Tais módulos são listados a seguir:

- Módulo **SPARKData** – Reúne definições de variáveis geométricas e de especificação do SPARK-NC, além de dicionários de características de materiais utilizados nos componentes do núcleo e de temperaturas iniciais desses componentes. Em suma, é o repositório de dados sobre o reator, onde as classes **CoreAssemblies**, **CoreMaterials** e **ReactorCore** buscam a maioria dos dados necessários para a instanciação de seus respectivos objetos.
- Módulo **SPARKThermal** – Reúne variáveis e funções necessárias ao estágio de ajuste termohidráulico.
- Módulo **SPARKDepletion** - Reúne variáveis e funções necessárias ao estágio de depleção.
- Módulo **SPARKUtilities** - É o módulo que contém as funções e classes necessárias para a criação completa do núcleo (as classes descritas acima estão inseridas nesse módulo), além de outras funções de utilidade geral que não se enquadram em nenhum dos módulos anteriores.

3.1.2.1 – A classe **CoreAssemblies**

A classe **CoreAssemblies** é a classe que cria todos os 8 tipos os elementos (combustíveis interno/médio/externo, controle primário e secundário, SCRAM, refletor e de blindagem). Além disso, ela cria um nono “elemento” que tem o mesmo formato prismático hexagonal dos elementos do reator mas é preenchido puramente por fluido refrigerante (LBE). A razão para a existência de tal elemento será explicada abaixo, quando o algoritmo para a montagem do reticulado hexagonal do núcleo for mostrada.

Conforme já mencionado, o OpenMC utiliza, a exemplo de outros códigos de Monte Carlo, métodos de geometria construtiva para a criação dos componentes do núcleo. Desta forma, cria-se objetos geometricamente complexos através da definição de superfícies relativamente simples e da

conjunção de regiões exteriores (+) ou interiores (-) de tais superfícies por meio de operadores de união (&), interseção (!) e diferença (!). Por exemplo, para a construção das regiões geométricas de um cerne de uma vareta de altura H e raio R com o eixo na direção z , isto pode ser feito da seguinte maneira no OpenMC (as unidades são em centímetros) :

```
import openmc

p1 = openmc.Zplane(z0=0)
p2 = openmc.Zplane(z0=H)
c = openmc.Zcylinder(r=R)
reg_cerne_vareta = -c & +p1 & -p2
```

Caso o usuário deseje colocar um revestimento cilíndrico de espessura r neste cerne e tampas superior e inferior de altura h cada uma, ele adicionará as seguintes linhas, além daquelas mostradas acima:

```
c2 = openmc.Zcylinder(r=R+r)
t1 = openmc.Zplane(z0=-h)
t2 = openmc.Zplane(z0=H+h)

reg_revestimento = +c & -c2 & +p1 & -p2
reg_tampa_sup = -c2 & +p2 & -t2
reg_tampa_inf = -c2 & -p1 & +t1
```

Pelos exemplos acima, nota-se que o usuário pode estender sem dificuldade os exemplos acima para a criação das diversas regiões geométricas de uma vareta (*gap*, refletores, *plenum*, etc.) conforme as especificações mostradas em 3.1.1 para todos os elementos do núcleo. Porém, para se atribuir características físicas (temperatura, densidade) e de composição material a esta região, ela deve ser associada com uma célula. Por exemplo, para a região correspondente ao cerne da vareta, deve-se fazer:

```
cerne_vareta = openmc.Cell(cell_ID = 1, region = reg_cerne_vareta)
```

Aqui, `region` recebe a região definida anteriormente e `cell_ID` é um número inteiro único que identifica essa célula no código (se esse número não for passado como argumento, ele será atribuído sequencialmente pelo OpenMC). Criada a célula, pode-se definir os materiais de acordo com a composição elementar, isotópica e densidade, atribuindo-a à célula desejada. Por exemplo, ao preencher a célula correspondente ao cerne da vareta com UO_2 enriquecido em 5% por fração atômica, faz-se:

```
uo2 = openmc.Material(material_ID = 1, name = "UO2")
```



```
uo2.add_nuclide('U235', 0.05)
uo2.add_nuclide('U238', 0.95) # UO2 enriquecido em 5%
uo2.add_nuclide('O16', 2.0)
uo2.set_density('g/cm3', 10.0)
cerne_vareta.fill = uo2
```

Para agregar todas as células formando um único objeto correspondente a uma vareta combustível revestida, cria-se um “universo” que contém todas as células já definidas:

```
vareta_revestida = openmc.Universe(cells=[cerne_vareta, revestimento, tampa_sup,
tampa_inf,])
```

O universo acima pode ser usado para preencher uma outra célula (que por sua vez pode fazer parte de um outro universo), ou ser replicado num reticulado hexagonal ou quadrado. Um código OpenMC pode ter vários universos, mas deve ter no mínimo um (que é denominado o universo principal).

Dado o código acima, pode-se inferir uma maneira de subdividir uma região no eixo vertical, criando uma célula distinta para cada subdivisão. Para a região “cerne da vareta”, por exemplo, pode-se criar um *loop Python* que “fatia” a região gradualmente em N partes ao longo do eixo z e atribui cada uma dessas partes a uma célula:

```
c = openmc.Zcylinder(r = R)
pinf = openmc.Zplane(z0 = 0) #plano inferior da fatia

for i in range(0,N): # i vai de 0 a N-1
    psup = openmc.Zplane(z0 = (i+1)*(H/N)) #plano superior da fatia
    reg_fatia = -c & +pinf & -psup
    fatias[i] = openmc.Cell(cell_ID = i, region = reg_fatia) # fatias é um vetor de contendo N
    células
    pinf = psup
```

Deve ser notado que foram criadas N células cuja região corresponde a uma fatia (seccionada no plano xy) da região anterior, com cada uma dessas células guardada em uma posição de um vetor. Desta maneira, ela pode ser passada como argumento na criação de um universo, juntamente com outras células, e esse universo conterá a definição de um cerne de vareta, exatamente como a anterior, mas sendo formada por um maior número de células. Visto que agora tem-se N células, cada uma delas pode receber uma temperatura diferente e um material contendo uma densidade diferente; no OpenMC, apenas materiais podem ter uma densidade atribuída – para

células isto não é permitido – e portanto para se variar a densidade criam-se materiais diferentes com a mesma composição isotópica mas densidades distintas e atribui-se esses materiais às diferentes células. Assim, consegue-se obter uma variabilidade de temperatura e densidade no eixo vertical de um dado componente (no caso, o cerne da vareta), com esta “granularidade” sendo determinada pelo número de células criadas.

Com as definições acima, pode-se criar o fluido refrigerante que circunda uma vareta, basta definir a região, subdividi-la a contento e então inseri-la no mesmo universo que contém a vareta (lembrando que o raio total da vareta é $R + r$):

```
c2 = openmc.Zcylinder(r = R+r)

reg_LBE = +c2 # região para fora da vareta
LBE = openmc.Cell(region = reg_LBE)

vareta_e_LBE = openmc.Universe(cells = [cerne_vareta, revestimento, tampa_sup,
tampa_inf, LBE, ])
```

A partir da criação do universo correspondente a uma vareta/fluido refrigerante circundante, com o número de componentes, o número de fatias de cada componente e o material de cada fatia definidos pelo usuário, pode-se replicar este universo em um reticulado hexagonal, de modo a criar um arranjo de varetas como o interior dos elementos mostrados na Subseção 3.1.1. No OpenMC, cria-se um objeto correspondente ao reticulado, passando como argumento três parâmetros: o centro do reticulado (isto é, o ponto central dos anéis concêntricos que caracterizam um reticulado hexagonal), o *pitch* do reticulado (o *pitch* pode ter um ou dois valores, dependendo do reticulado fazer sua repetição em uma ou duas direções) e o universo que preencherá a parte externa do reticulado (isto é, a região que é contígua ao anel mais externo do reticulado). Então, para um reticulado centrado em (0,0) no plano xy com *pitch* de 1.24 cm e cuja parte externa é preenchida com um universo contendo apenas fluido refrigerante, tem-se:

```
reticulado = openmc.HexLattice()

reticulado.center = (0,0)
reticulado.pitch = (1.24,)
reticulado.outer = universo_LBE #esse universo é definido em outra parte do código
```

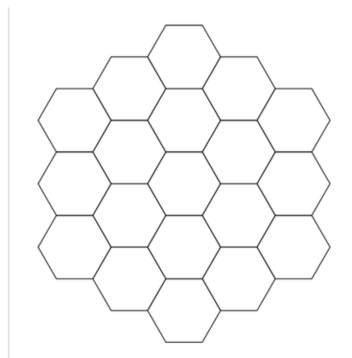
A criação do reticulado define uma estrutura de regiões delimitadas por um hexágono equilátero cujo lado é univocamente determinado pelo *pitch* (ver Figura 3.7 (b)), conectadas e indexadas

segundo um sistema específico: trata-se de uma dupla de índices (a, b) . O índice a corresponde à localização do elemento hexagonal no anel do hexágono, indo de $a = 0$ no anel mais externo até $n-1$, onde n é o número de anéis do reticulado. O índice b cresce de $b = 0$ no elemento na parte superior extrema de um dado anel e cresce no sentido horário. Como exemplo, para um reticulado com 3 anéis (19 elementos hexagonais), a indexação é dada conforme a Figura 3.7 (a).

Figura 3.7 – Indexação de um reticulado hexagonal de 19 elementos (a) e figura mostrando tal reticulado (b).

(a)

		(0, 0)		
	(0, 11)	(1, 0)	(0, 1)	
(0, 10)	(1, 5)	(2, 0)	(1, 1)	(0, 2)
(0, 9)	(1, 4)	(1, 2)	(0, 3)	
(0, 8)	(1, 3)	(0, 4)		
	(0, 7)	(0, 5)		
	(0, 6)			



(b)

É importante mencionar que, por *default*, o reticulado está com “orientação em y”, o que significa que seus dois lados retos estão orientados paralelos ao eixo y (a mesma definição de orientação é válida também para apenas uma figura hexagonal ao invés de um reticulado). Caso seja desejada uma “orientação em x”, deve-se ter no código

```
reticulado.orientation = "x"
```

A cada um desses espaços hexagonais mostrados na Figura 3.7, atribui-se um universo por meio de sua indexação, podendo ser universos distintos – o que seria correspondente a varetas distintas dentro de um elemento – ou o mesmo universo. O espaço hexagonal definido delimita o universo que é atribuído a este espaço (note que o universo definido até então, que abrange a vareta e o LBE circundante, tem extensão infinita no espaço), mas essa delimitação é apenas no plano xy , pois o reticulado é em duas dimensões. Deste modo, os universos inseridos no reticulado ainda permanecem de extensão infinita nas direções z positiva e negativa e, além disso, o universo externo ao reticulado (*outer universe*) pode, a depender de sua definição, se estender infinitamente. Porém, isto será remediado quando reticulado for inserido na célula correspondente ao elemento de varetas, que é um prisma de extensão finita e portanto limita o que nele for inserido.

A atribuição é feita de acordo com a indexação (a, b) da seguinte forma: Para cada anel, cria-se uma variável diferente que é um vetor no *Python*; este vetor deve ser um vetor de universos, onde

o universo da posição b desse anel deve estar na posição b do vetor. Por exemplo, suponha que deseja-se preencher o reticulado de 3 anéis com dois universos `univ_1` e `univ_2` com a seguinte configuração:

- Anel mais externo ($a = 0$): os três primeiros elementos contando a partir daquele que está na parte superior extrema do anel e seguindo no sentido horário (ou seja $b=0$ até $b=2$) sendo preenchidos com universo 1 e o restante com universo 2;
- Anel intermediário ($a = 1$): totalmente preenchido com o universo 1;
- Anel mais interno ($a = 2$): o único elemento deve ser preenchido com o universo 2.

Deste modo, tem-se no código:

```
#vetor com posições 0 a 2 com univ_1 e 3 a 11 com univ_2
anel_externo = 3*[univ_1] + 9*[univ_2]
anel_intermediario = 6*[univ_1] #vetor onde as 6 posições estão com univ_1
anel_interno = [univ_2] #vetor de uma posição com univ_2
```

Posteriormente, cada variável de anel é inserido em um vetor onde a posição no vetor é a posição a da indexação (a, b) , sendo esse vetor atribuído à variável `universes` do objeto correspondente ao reticulado:

```
reticulado.universes = [anel_externo, anel_intermediário, anel_interno]
```

Tal método pode ser estendido para um reticulado com qualquer número de anéis, sendo preenchido por qualquer número de universos. Com a criação do reticulado contendo o arranjo de varetas, é necessário inseri-lo em um elemento de varetas de seção transversal hexagonal, conforme mostrado em 3.1.1. Para este fim, o OpenMC oferece uma função que já cria diretamente um “cilindro” de seção transversal hexagonal equilátera. Assim, a região interna de um prisma hexagonal com eixo em z , altura h , orientação do hexágono em “ x ” e lado L é dada por

```
superficie_hex = openmc.model.HexagonalPrism(edge_length=L, orientation='x')
p1 = openmc.Zplane(z0=0)
p2 = openmc.Zplane(z0=h)
reg_prism_hex = -superficie_hex & +p1 & -p2
```

A região prismática criada é atribuída então a uma célula. Conforme mencionado, células podem ser preenchidas por materiais, mas também podem ser preenchidas com reticulados como aqueles discutidos acima. Portanto, para a criação da parte interna de um elemento de varetas, pode-se inserir o reticulado hexagonal dentro da célula correspondente ao prisma hexagonal adicionando-se a seguinte linha ao código acima:

```
parte_interna_elem = openmc.Cell(region=reg_prism_hex, fill=reticulado)
```

Conforme mencionado anteriormente, a inserção do reticulado dentro de uma célula prismática de volume finito, como é o caso de `parte_interna_elem`, efetivamente limita o reticulado em todas as direções. Após a criação da parte interna do elemento de varetas, cria-se o revestimento externo do mesmo. Como a superfície prismática da parede interna do revestimento já encontra-se criada (`superficie_hex`), basta apenas criar a superfície da parede externa, que terá um lado $L + \Delta$, onde Δ é calculado de modo que a espessura do revestimento do elemento de varetas seja conforme a desejada:

```
superficie_hex_ext = openmc.model.HexagonalPrism(edge_length=L+Delta,orientation='x')
```

Com isto, pode-se criar a célula correspondente ao revestimento do elemento de varetas da seguinte forma:

```
revest_elem = openmc.Cell(region = +superficie_hex & -superficie_hex_ext & +p1 & -p2)
```

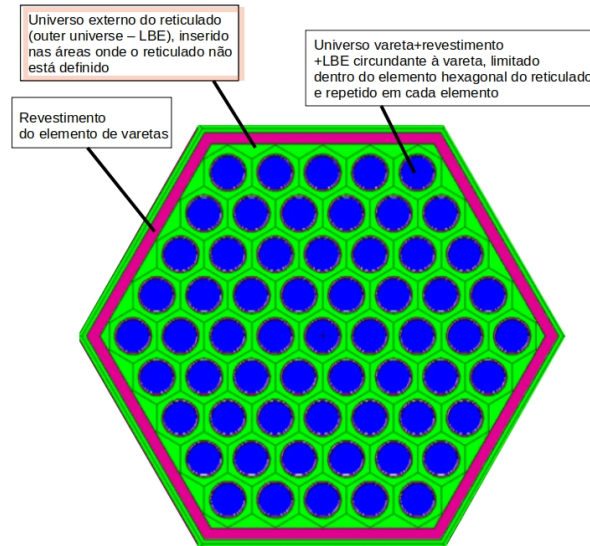
A célula correspondente ao revestimento também pode ser subdividida em várias outras células, da mesma maneira que já foi mencionada anteriormente, para que se tenha uma maior granularidade das propriedades do revestimento ao longo do eixo z.

Finalmente, pode-se reunir as células correspondentes ao revestimento e a parte interna do elemento de varetas em um único universo, que compreende toda a geometria do elemento de varetas:

```
elemento_varetas = openmc.Universe(cells=[revest_elem, parte_interna_elem,])
```

O resultado final de todo o código acima seria uma geometria cujo corte transversal é similar ao mostrado na Figura 3.8 (a figura mostra um elemento de varetas com um reticulado de 5 anéis – 61 elementos).

Figura 3.8 – Elemento de varetas composto por um reticulado hexagonal de 5 anéis (61 elementos).



A classe **CoreAssemblies** é, em resumo, uma aplicação mais convoluída das funções e definições mostradas acima, devido à versatilidade na definição da mesma: A partir dos argumentos de entrada, pode-se criar qualquer um dos oito elementos utilizados na montagem do núcleo (com todos os componentes associados a um dado tipo: combustível, revestimento, *gap*, *plenum*, refletores das varetas, LBE circundante, revestimento do elemento, etc.), além do já mencionado elemento que consiste de um prisma hexagonal preenchido apenas com fluido refrigerante (LBE). Portanto, grande parte do código da classe é dedicado às diferenças de especificação e componentes das varetas e número de anéis dos reticulados hexagonais (os elementos combustíveis possuem cinco anéis, enquanto os demais possuem apenas três), entre outros parâmetros. Além disto, para os elementos de controle/SCRAM, pode-se passar como argumento o percentual de inserção do elemento no núcleo; neste caso, o código cria a parte interna do elemento com todos os parâmetros geométricos relacionados ao eixo z deslocados por um valor relacionado ao percentual, inserindo uma célula prismática hexagonal preenchida unicamente por LBE na região deslocada.

É importante mencionar que, ao se instanciar a classe durante o código, é passado como argumento tão somente o tipo de elemento que se quer construir, o conjunto de materiais necessários para o preenchimento das células (que provém de um objeto da classe **CoreMaterials**) e, caso seja um elemento de controle/SCRAM, o percentual de inserção. Todos os outros parâmetros geométricos para a construção são obtidos diretamente do módulo **SPARKData** (tais parâmetros

tornam-se variáveis de ambiente quando o módulo é importado e são acessados durante a instanciamento da classe **CoreAssemblies** para a construção do elemento. Portanto, caso se deseje alterar as dimensões dos parâmetros para a criação de um determinado elemento, deve-se alterar as variáveis relacionadas nesse módulo antes de carregá-lo no código.

Outro conjunto de dados que é derivado diretamente do módulo **SPARKData** são as distribuições de temperaturas iniciais para cada uma das subdivisões de cada um dos componentes do elemento. Existe uma função definida no módulo **SPARKUtilities** que é chamada de **sliceCells** e tem por objetivo criar os componentes do elemento subdivididos em várias células de uma maneira similar àquela mencionada nos parágrafos acima. O número de subdivisões de um dado componente é feito com base no tamanho do vetor de temperaturas iniciais definidas em **SPARKData** para aquele componente, e a temperatura de cada subdivisão é atribuída à célula associada na medida que as mesmas são criadas pela função. Deste modo, a granularidade do modelo do núcleo, isto é, a quantidade de subdivisões das células, é controlada neste processo: quando se deseja uma maior ou menor granularidade, aumenta-se ou diminui-se o tamanho dos vetores de temperaturas iniciais associados a cada componente por extrapolação ou interpolação, induzindo portanto uma maior ou menor subdivisão desses componentes.

Reitera-se que as temperaturas mencionadas no parágrafo acima são definidas apenas para a criação do objeto núcleo inicial, pois elas são alteradas em ambos os estágios de cálculo: No ajuste termohidráulico a distribuição de temperatura de todos os componentes do núcleo são modificadas até se atingir um valor de estado estacionário e no estágio de depleção a distribuição de temperatura dos componentes do núcleo é substituída por aquela herdada do estágio de ajuste termohidráulico imediatamente anterior.

Um resumo do processo de criação de um elemento do núcleo pela classe **CoreAssemblies** é dada abaixo:

- Ao se instanciar a classe, passa-se como argumento o tipo de elemento a ser criado, além dos materiais para o preenchimento das células (oriundo de um objeto da classe **CoreMaterials**) e o percentual de inserção, caso seja um elemento de controle/SCRAM. Também é passada na instanciamento da classe o grau de granularidade das subdivisões de cada componente.
- A partir do tipo de elemento que se deseja criar, a classe carrega as especificações geométricas necessárias no módulo **SPARKData**, e inicia a criação dos componentes associadas à vareta daquele elemento (combustível/absorvedor, refletores, *plenum*, fluido

refrigerante circundante à vareta etc.). Cada um desses componentes são subdivididos em diversas células pela função `sliceCells` de acordo com o tamanho do vetor de distribuição de temperaturas iniciais atinente àquele componente, com as temperaturas e materiais sendo atribuídas às células à medida que as mesmas são criadas. Caso um percentual de inserção for definido, os parâmetros geométricos obtidos em **SPARKData** são deslocados na direção *z* de acordo com o valor do percentual, preenchendo-se a parte deslocada com uma célula preenchida por fluido refrigerante (LBE).

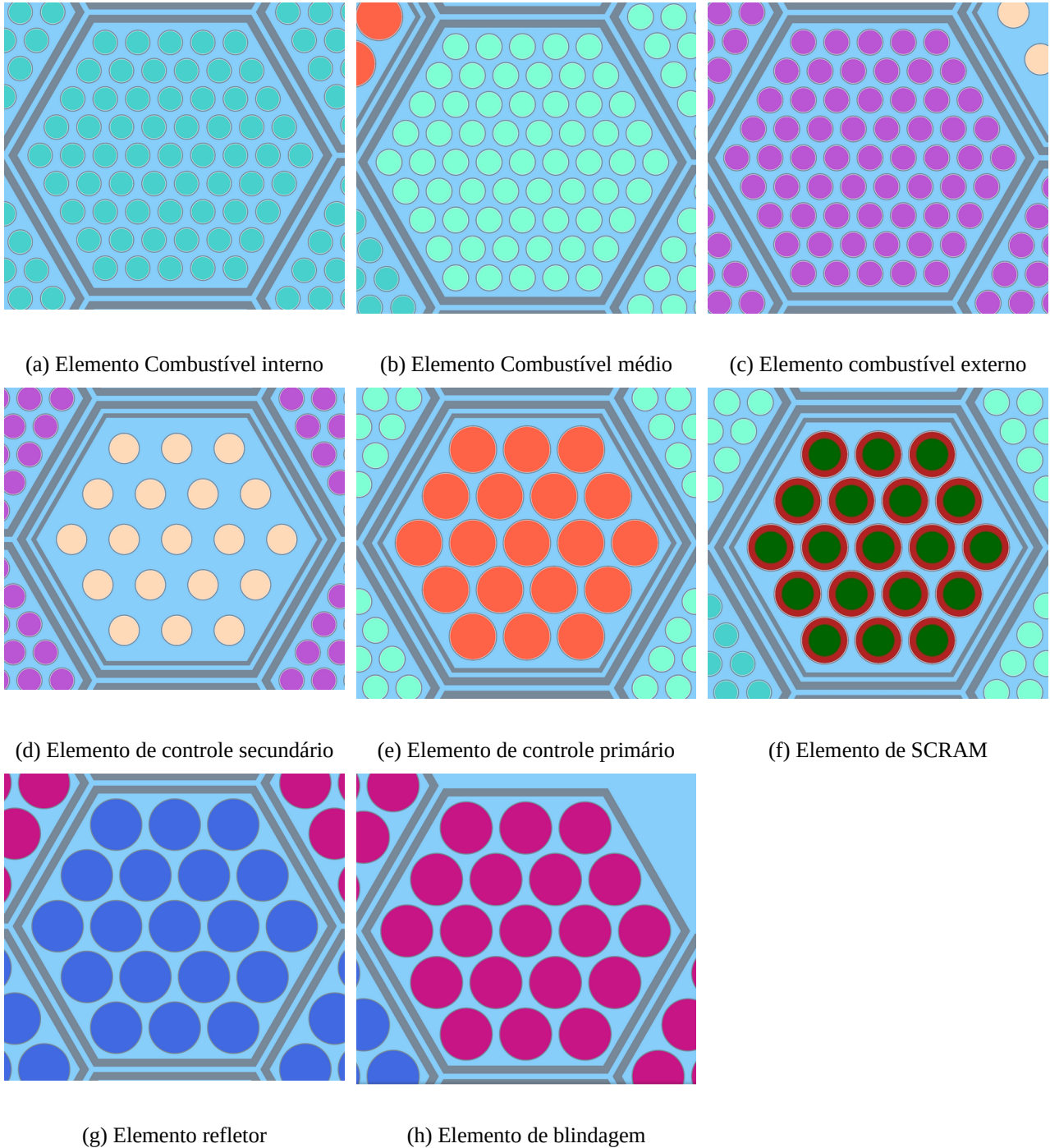
- A partir dos componentes das varetas, cria-se o universo correspondente à vareta daquele elemento em conjunção com o fluido refrigerante que o circunda, e então define-se o reticulado hexagonal a partir de dados como o número de anéis do reticulado e o *pitch* para aquele tipo de elemento (que também é obtido no módulo **SPARKData**), além de se definir o universo externo ao reticulado, preenchendo-o com LBE.
- Cria-se uma célula prismática hexagonal de acordo com especificações geométricas advindas do módulo **SPARKData** e preenche-se o mesmo com o reticulado definido na etapa anterior. Este prisma preenchido define a parte interna do elemento. Cria-se então o revestimento externo do elemento, que também é subdividido em várias células e tem seu material e sua distribuição de temperatura inicial devidamente atribuída a essas células. Conforme já mencionado anteriormente, os elementos de controle/SCRAM são mais complexos: O elemento encontra-se inserido em uma casca prismática fixa que é também tem seção transversal hexagonal e é concêntrica ao elemento, com o elemento deslizando no eixo *z* dentro dessa casca e uma camada de fluido refrigerante entre o elemento e a estrutura (ver Figura 3.9 (d)-(f) para a seção transversal dos elementos de controle e SCRAM). Deste modo, deve ser providenciada a criação de três componentes: o elemento, a estrutura e a camada de LBE, com suas respectivas subdivisões e atribuições de materiais e temperaturas iniciais.
- Todos os conjuntos de células são congregados em um único universo, que representa o elemento como um todo. Esse universo é associado a um atributo do objeto, que pode ser acessado quando necessário no código.

Conforme a descrição da última etapa do processo descrito acima, o resultado da instanciação de um objeto da classe **CoreAssemblies** associado a um dado tipo de elemento do núcleo é a criação de um universo descrevendo o elemento. Por ser um universo, ele pode ser

inserido em um outro reticulado hexagonal, desta vez descrevendo a geometria do núcleo. Tal processo será mostrado em 3.1.2.3, quando a classe **ReactorCore** for discutida.

A Figura 3.9 (a)-(h) mostram uma seção transversal de cada tipo de elemento criado para o núcleo do SPARK-NC no OpenMC.

Figura 3.9 – Seção transversal de cada um dos elementos que compõem o SPARK-NC, criados no OpenMC.



3.1.2.2 - A classe **CoreMaterials**

A classe **CoreMaterials** cria um objeto que é, conforme já foi mencionado, um repositório de materiais para o preenchimento das células que integram os diversos componentes do núcleo. A sua estrutura é relativamente simples, baseando-se em uma função denominada **setMaterials** que encontra-se definida em **SPARKUtilities**. Em resumo, essa função recebe, dentre outros argumentos, um “dicionário” relacionado ao material que se deseja criar, além da densidade desejada para o material e a informação se o material é físsil ou não (tal informação é importante durante o estágio de depleção). Um dicionário no *Python* é uma espécie de vetor que pode ser indexado não só por números, como um vetor comum, mas também por uma *string* de texto. Os dicionários relacionados aos materiais encontram-se no módulo **SPARKData**, onde pode-se encontrar, por exemplo, o dicionário relacionado ao UO₂ enriquecido a 17%:

```
URANIUM_OXIDE_17 = {'elements': {'U' : 1.0, 'O16' : 2.0} , 'enrichment': {'U235' : 17.0}, 'density' : 10.939, 'density factor': 94.165, 'S(a,b)' : ['c_0_in_UO2', 'c_U_in_UO2']}
```

Neste caso, caso se deseje acessar a densidade do material (*density*), basta fazer:

```
In[1]: URANIUM_OXIDE_17['density']  
Out[1]: 10.939
```

Note que existem dicionários aninhados dentro de outros dicionários. Portanto, para acessar a fração atômica do Urânio no material, faz-se:

```
In[1]: URANIUM_OXIDE_17['elements']['U']  
Out[1]: 1.0
```

Durante a instânciação do objeto **CoreMaterials**, a função **setMaterials** é chamada para cada material criado para o preenchimento das células que integram os diversos componentes do núcleo. Deve ser mencionado que um único material pode ser utilizado para o preenchimento de várias células distintas (um exemplo são as várias células que subdividem um mesmo componente, que é composto por um mesmo material e possuem a mesma densidade), mas também existem componentes, como o fluido refrigerante, que exigem uma variabilidade maior de densidade e portanto exigem a criação de diversos materiais distintos.

Um resumo dos diversos materiais que compõem o núcleo do SPARK-NC, com suas especificações associadas, é dado na Tabela 3.5.

Tabela 3.5 – Resumo dos materiais que compõem o núcleo do SPARK-NC.

Material	Especificações/comentários
Dióxido de Urânio (UO₂)	<p>Foram criados três materiais distintos, referentes às três regiões do núcleo (interna, média e externa), Deste modo, todos as células correspondentes à região ativa de cada vareta numa dada região é preenchida com o mesmo material. As densidades de cada um dos três materiais foram definidas pela fórmula mostrada a seguir, que relaciona a densidade em g/cm³ do UO₂ com a fração de enriquecimento ϵ, considerando que a densidade teórica do UO₂ com o urânio contendo apenas o isótopo ²³⁸U é 10.96 g/cm³ e aquele contendo apenas o isótopo ²³⁵U é de 10.838 g/cm³ [5]:</p> $\rho(\epsilon) = 10.96(1 - \epsilon) + 10.838\epsilon.$ <p>Assim, os seguintes valores de densidade teórica dos materiais enriquecidos foram especificados:</p> $\rho_{interno} = \rho_{medio} = \rho(\epsilon = 0.17) = 10.939g/cm^3$ $\rho_{externo} = \rho(\epsilon = 0.1975) = 10.936g/cm^3$ <p>A densidade teórica assume que todo o volume é preenchido com os átomos do material; entretanto, considera-se que o combustível possui uma certa porosidade. Para este fim, é definido um fator menor que 1, que é multiplicado pela densidade teórica. Tal fator é encontrado ao se dividir a massa total do inventário de metal pesado especificada para o reator no início do ciclo de vida (5745 kg, ver Tabela 3.4) pela massa do mesmo inventário quando o mesmo é preenchido pelo combustível com a densidade teórica (o que pode ser obtido para um dado núcleo por um atributo do OpenMC chamado heavy_metal, ver Subseção 3.3.2). Deste modo, encontrou-se um fator de 0.94165, que foi aplicado em ambas as densidades resultando em</p> $\rho_{interno} = \rho_{medio} = 10,300, \rho_{externo} = 10,297.$ <p>A definição do material também emprega tabelas $S(\alpha, \beta)$ para seções de choque de átomos ligados para nêutrons incidentes em energias térmicas. As seguintes tabelas são utilizadas:</p> <ul style="list-style-type: none"> • Oxigênio presente no UO₂, • Urânio presente no UO₂. <p>As especificações acima podem ser observadas na definição do dicionário associado ao UO₂ (URANIUM_OXIDE_17), que é mostrado logo após o primeiro parágrafo da Subseção 3.1.2.2 .</p>
Chumbo-Bismuto Eutético (Lead Bismuth Eutectic LBE)	<p>Os autores não mencionam a composição em fração atômica da liga de Chumbo-Bismuto Eutético utilizada. Diversas composições percentuais de Chumbo-Bismuto são encontradas na literatura, mas a composição utilizada – 44.5% Pb e 55.5% Bi em fração de peso, ou 44.7% Pb e 55.3% Bi em fração atômica – é a encontrada na referência [6], que mostra as curvas relacionando diversas características físicas do LBE com a temperatura. Em tal referência, a relação densidade-temperatura recomendada é dada por:</p> $\rho = 11.096 - 1.3236(T/1000)g/cm^3, \text{ onde a temperatura } T \text{ é dada em Kelvin.}$ <p>Assim, a densidade inicial do material é determinada de acordo com a temperatura inicial da célula que é preenchida pelo material, em conjunção com a relação dada conforme acima.</p>
Aço T91	<p>Conforme já mencionado, os autores não indicam em [1] a composição do T91 utilizado, e portanto é utilizada a referência [2], que discute sobre a interação físico-química do LBE com metais como o T91, fornecendo a composição em fração percentual de peso dos diversos elementos empregados na liga. A composição do Aço T91 utilizada no modelo do SPARK-NC, que é baseada na referência supramencionada, é dada na Tabela 3.2. Deve ser mencionado que, devido ao fato da maioria dos átomos da liga serem de Ferro (> 88%) e o isótopo Fe-56 ser o mais abundante (~91.7%), o modelo do</p>

Material	Especificações/comentários
	material incorpora uma tabela $S(\alpha, \beta)$ relativa ao Ferro-56 para o ajuste de seções de choque em energias térmicas, que está presente na biblioteca ENDF-VII.0 (que é a biblioteca de seções de choque utilizada pelo SARAX conforme [1], e é a biblioteca que é empregada também na simulação, para fins de consistência). A densidade do aço T91 é praticamente padronizado na literatura técnico-científica, estando em torno de 7.7 g/cm^3 , e portanto este valor é empregado no modelo.
Material Absorvedor - Carbetto de Boro (B_4C)	O material foi modelado com uma densidade de 2.52 g/cm^3 . Visto que em [1] especifica-se que o Carbetto de Boro utilizado nos elementos de blindagem possui enriquecimento natural em relação ao isótopo ^{10}B , o material absorvedor desses elementos é composto por B_4C com 19.9% de ^{10}B . As varetas de SCRAM são modeladas com o dobro de enriquecimento, 40%.
Tungstênio	Modelado com a densidade padrão encontrada na literatura, 19.25 g/cm^3 .
Hélio	O Hélio utilizado para o preenchimento do <i>gap</i> e do <i>plenum</i> das varetas foi definido com a densidade de temperatura ambiente, 0.0001786 g/cm^3 . Como há pouca variabilidade em uma faixa entre $330 \text{ }^\circ\text{C}$ e $600 \text{ }^\circ\text{C}$ (aproximadamente a faixa de temperatura de operação do SPARK-NC), tal valor foi mantido inalterado durante as simulações.

Visto que todos os materiais, com exceção do LBE, têm a sua densidade imutável durante as simulações, apenas uma instância de cada um desses materiais é criada por **setMaterials**, com as suas respectivas densidades conforme a Tabela 3.5. Portanto, todas as células não-LBE serão preenchidas por esse único objeto correspondente material de densidade fixa, sendo a temperatura do material naquela célula definida ou alterada em tempo de execução através de uma atribuição de temperatura diretamente na célula.

Quanto ao LBE, o objeto **CoreMaterials** cria vários materiais com a mesma composição isotópica do LBE, com o número de materiais criados a depender da granularidade do modelo (mais refinado para o ajuste termohidráulico ou menos refinado para o cálculo de depleção):

- Para o modelo de núcleo mais refinado, cria-se um material distinto para cada região de LBE que circunda uma dada vareta de elemento (o que seria a região hexagonal onde está inserida uma dada vareta no universo correspondente ao elemento, ver Figura 3.8). Deve ser lembrado que esse LBE circundante a uma vareta pode, como qualquer componente da vareta, ser subdividido longitudinalmente em várias células, o que resultaria em várias “fatias” do LBE na direção do eixo z . Deste modo, como há apenas um material por região de LBE, tal material é atribuído a todas às subdivisões dessa região, sendo a sua densidade calculada conforme a relação densidade-temperatura para o LBE na Tabela 3.5, onde a temperatura é dada pela média da temperatura de todas as subdivisões.
- Para o modelo menos refinado, é definido apenas um material para a região conjunta do LBE que circunda todas as varetas de um dado elemento no núcleo. Deste modo, a granularidade na especificação da densidade é no nível do reticulado hexagonal do núcleo,

ao contrário do modelo mais refinado, que é no nível do reticulado hexagonal interior a cada elemento de varetas. Por ter apenas um único material por elemento, a densidade de tal material é dada pela média da temperatura de todas as subdivisões das regiões de LBE circundantes à todas as varetas do elemento em questão, com a relação temperatura-densidade para o LBE conforme dada na Tabela 3.5.

- Para elementos que são desprovidos de varetas, como os “elementos” compostos puramente por fluido refrigerante (ver o primeiro parágrafo de 3.1.2.1) ou pela região correspondente a um elemento de controle/SCRAM totalmente retirado do núcleo, é definido um único material LBE, para todo o elemento, independente do modelo ser de maior ou menor refinamento. Neste caso, a densidade também é dada pela média das temperaturas das células correspondentes às subdivisões dessas regiões, através da supracitada relação densidade-temperatura.

Antes da criação do núcleo pela classe **ReactorCore**, uma instância da classe **CoreMaterials** é criada, resultando em um objeto que é efetivamente o repositório de materiais que compõem o núcleo. As densidades dos materiais não-LBE são atribuídas conforme especificação, enquanto as densidades de cada material LBE é definida de acordo com a temperatura com que as células de LBE são inicializadas, segundo a relação densidade-temperatura mostrada na Tabela 3.5. Ambos os dados (densidades dos materiais não-LBE e temperaturas iniciais de todos os componentes de um dado elemento, incluindo regiões de LBE), além das composições isotópicas e graus de enriquecimento de todos os materiais do núcleo, devem ser definidos no módulo **SPARKData** antes da execução do código.

3.1.2.3 - A classe **ReactorCore**

A classe **ReactorCore** define o objeto que representa toda a geometria do núcleo, em conjunção com os materiais que o compõem. A classe **ReactorCore** recebe, no momento de sua instanciação, os seguintes argumentos:

- Uma variável que indica a granularidade do núcleo;
- O repositório de materiais que compõem o núcleo, que é um objeto da classe **CoreMaterials**;

- O percentual de inserção dos elementos de controle/SCRAM com o qual se deseja criar o núcleo.

O processo de criação do núcleo dá-se em três etapas: primeiramente, cria-se objetos da classe **CoreAssemblies** representando todos os nove tipos de elementos: os oito tipos de elementos de vareta e o elemento que consiste apenas de fluido refrigerante. Deve ser notado que é criado apenas um único objeto para cada tipo, que serão posteriormente replicados em um reticulado hexagonal que representa o núcleo. Para cada elemento que é criado, são fornecidos todos os materiais necessários através do repositório único de materiais, além do grau de refinamento do elemento e o percentual de inserção do elemento, caso seja do tipo controle/SCRAM. O processo de criação de um objeto **CoreAssemblies** representando um elemento é mostrado na Subseção 3.1.2.1.

Numa segunda etapa, o reticulado hexagonal representando o núcleo é criado. O código extrai a informação de um vetor bidimensional definido no módulo **SPARKData** chamado **RINGS_MAP**. A primeira posição desse vetor denota o anel do reticulado, com o anel mais interno indexado com o valor 0, com o índice crescendo do anel mais externo para o mais interno (ou seja, o índice a na indexação (a, b) da Figura 3.7), enquanto a segunda posição aloca dois vetores: o primeiro mostra quantos elementos de um dado tipo são inseridos sequencialmente no anel, a partir da posição do extremo norte do anel, prosseguindo-se no sentido horário (ou seja, a posição $b=0$ na indexação (a, b) da Figura 3.7), enquanto o segundo vetor mostra o tipo de elemento que será sequencialmente inserido. Por exemplo, as entradas correspondentes aos dois anéis mais externos do reator são dadas por:

```
RINGS_MAP[0][0] = [72]
RINGS_MAP[0][1] = ['COOLANT']

RINGS_MAP[1][0] = [2,8,3,8,3,8,3,8,3,8,3,8,1]
RINGS_MAP[1][1] = ['COOLANT', 'SHIELD', 'COOLANT', 'SHIELD', 'COOLANT',
'SHIELD', 'COOLANT', 'SHIELD', 'COOLANT', 'SHIELD', 'COOLANT', 'SHIELD', 'COOLANT']
```

As linhas acima carregam a seguinte informação: todos as 72 regiões hexagonais do anel mais externo do reticulado são preenchidas pelo elemento composto puramente por LBE, enquanto o anel concêntrico imediatamente interior ao anel mais externo possui as primeiras duas regiões preenchidas por um “elemento puramente LBE” (começando-se da região hexagonal mais ao norte do anel e seguindo em sentido horário), com as oito regiões na sequência preenchidas por elementos

de blindagem, as três regiões seguintes preenchidas por elementos LBE, e assim por diante, até que todos as 66 regiões desse anel sejam preenchidas.

Incidentalmente, o código acima mostra a motivação para se definir um elemento que é puramente composto por LBE: Observa-se, pela Figura 3.3, que o anel mais externo do reticulado no núcleo do SPARK não é totalmente preenchido por elementos de blindagem: os seis “cantos” do anel não são compostos por nenhum elemento de vareta, sendo tais espaços preenchidos por fluido refrigerante. Assim, a criação de um universo que representa um “elemento” preenchido puramente por LBE torna-se necessário para ser inserido nessas posições do reticulado hexagonal.

Nota-se também pela Figura 3.3 que entre o reticulado contendo os elementos e a parede do vaso do reator há um espaço preenchido por fluido refrigerante. Tal preenchimento poderia ser obtido ao se criar um universo composto unicamente por LBE e defini-lo como o universo externo ao reticulado (ver a discussão sobre a variável *outer universe* na Subseção 3.1.2.1). Porém, esse universo ocuparia uma região muito extensa do núcleo, sendo que apenas uma temperatura/densidade poderia ser atribuído ao mesmo. Portanto, de modo a manter a possibilidade de se atribuir diferentes temperaturas/densidade do LBE em diferentes pontos desta região, optou-se por incluir um anel, externo a todos os outros, que é composto exclusivamente por elementos-LBE e ocupa totalmente a região entre a parede do vaso do reator e os anéis de elementos de varetas, conforme pode ser observado na estrutura do vetor `RINGS_MAP` mostrada acima.

A descrição completa do vetor `RINGS_MAP`, conforme aparece definida no módulo **SPARKData**, é mostrado abaixo deste parágrafo. Tal código pode ser comparado com a Figura 3.3, que provém do artigo original sobre o SPARK-NC [1] e a Figura 3.12, que mostra um corte transversal do modelo do SPARK-NC produzido no OpenMC; pode ser observado que, nestas figuras, a distribuição dos elementos em cada um dos anéis, originando-se no ponto mais ao norte do anel e prosseguindo no sentido horário, está de acordo com a indexação na Figura 3.7. Desta maneira, caso se deseje rearranjar a distribuição (incluindo a quantidade e o número de anéis) dos elementos no núcleo, basta alterar diretamente `RINGS_MAP` no módulo **SPARKData** antes da execução do código.

```
RINGS_MAP[0][0] = [72]
RINGS_MAP[0][1] = ['COOLANT']

RINGS_MAP[1][0] = [2,8,3,8,3,8,3,8,3,8,3,8,1]
RINGS_MAP[1][1] = ['COOLANT', 'SHIELD', 'COOLANT', 'SHIELD', 'COOLANT',
'SHIELD', 'COOLANT', 'SHIELD', 'COOLANT', 'SHIELD', 'COOLANT', 'SHIELD', 'COOLANT']

RINGS_MAP[2][0] = [2,7,3,7,3,7,3,7,3,7,3,7,1]
RINGS_MAP[2][1] = ['SHIELD', 'REFLECTOR', 'SHIELD', 'REFLECTOR', 'SHIELD', 'REFLECTOR',
```

```

'SHIELD', 'REFLECTOR', 'SHIELD', 'REFLECTOR', 'SHIELD', 'REFLECTOR', 'SHIELD']

RINGS_MAP[3][0] = [54]
RINGS_MAP[3][1] = ['REFLECTOR']

RINGS_MAP[4][0] = [2,5,3,5,3,5,3,5,3,5,3,5,1]
RINGS_MAP[4][1] = ['REFLECTOR', 'OUTER FUEL', 'REFLECTOR', 'OUTER FUEL', 'REFLECTOR',
'OUTER FUEL', 'REFLECTOR', 'OUTER FUEL', 'REFLECTOR', 'OUTER FUEL', 'REFLECTOR', 'OUTER
FUEL', 'REFLECTOR']

RINGS_MAP[5][0] = [4,1,13,1,13,1,9]
RINGS_MAP[5][1] = ['OUTER FUEL', 'SECONDARY CONTROL', 'OUTER FUEL', 'SECONDARY
CONTROL', 'OUTER FUEL', 'SECONDARY CONTROL', 'OUTER FUEL']

RINGS_MAP[6][0] = [9,1,11,1,11,1,2]
RINGS_MAP[6][1] = ['OUTER FUEL', 'PRIMARY CONTROL', 'OUTER FUEL', 'PRIMARY CONTROL', 'OUTER
FUEL', 'PRIMARY CONTROL', 'OUTER FUEL']

RINGS_MAP[7][0] = [30]
RINGS_MAP[7][1] = ['OUTER FUEL']

RINGS_MAP[8][0] = [24]
RINGS_MAP[8][1] = ['MID FUEL']

RINGS_MAP[9][0] = [1,2,1,2,1,2,1,2,1,2,1,2]
RINGS_MAP[9][1] = ['PRIMARY CONTROL', 'MID FUEL', 'SCRAM', 'MID FUEL', 'PRIMARY
CONTROL', 'MID FUEL', 'SCRAM', 'MID FUEL', 'PRIMARY CONTROL', 'MID FUEL', 'SCRAM', 'MID FUEL']

RINGS_MAP[10][0] = [12]
RINGS_MAP[10][1] = ['INNER FUEL']

RINGS_MAP[11][0] = [6]
RINGS_MAP[11][1] = ['INNER FUEL']

RINGS_MAP[12][0] = [1]
RINGS_MAP[12][1] = ['INNER FUEL']

```

Criado o universo correspondente ao reticulado hexagonal com todos os elementos do núcleo, deve-se inseri-lo em uma célula que representa a região interior do vaso do reator. Para este fim, cria-se uma célula cilíndrica cujo diâmetro é dado pelo diâmetro interno do vaso e a altura é igual à altura do núcleo (dados na Tabela 3.4). Após o preenchimento da célula com o reticulado, cria-se uma célula referente a uma casca cilíndrica de espessura referente à diferença entre o raio externo e interno do vaso e altura também igual à altura do núcleo, que é preenchida com aço T91, definindo a parede do vaso do reator.

Para a execução da simulação, deve-se especificar condições de contorno nas fronteiras da geometria do núcleo. Deste modo, impõe-se uma condição de contorno de vácuo na superfície cilíndrica correspondente à parte externa da parede do vaso do reator, além dos planos superior e inferior da célula cilíndrica que definem a região interior do vaso. Isto é feito de modo a terminar a história das partículas que atravessam tais superfícies, efetivamente emulando uma região idealizada de vácuo em torno do reator, onde não existam meios que possam emitir ou refletir partículas para o interior do núcleo.

Finalmente, faz-se a junção das células representando a parede do vaso do reator e a parte interna do vaso com os elementos sob um mesmo universo, que representa todo o núcleo e é o universo principal da simulação (tal universo é também denominado no OpenMC “universo pai” e nele estão contidos todos os outros universos.).

Criado o universo-pai que representa todo o núcleo, é possível acessar um dado universo que está em qualquer um dos dois níveis de reticulado hexagonal (nível de elementos no núcleo e nível de varetas/LBE em um dado elemento). No OpenMC, isto é feito através do sistema de coordenadas de números inteiros $(\hat{\alpha}, \hat{x})$: A Figura 3.10 mostra as indexações para um reticulado de 3 anéis (19 hexágonos) orientado em ‘y’. A partir dessa figura, é possível deduzir uma fórmula para as coordenadas cartesianas (x, y) do centro de qualquer hexágono de acordo com sua coordenada $(\hat{\alpha}, \hat{x})$:

$$\begin{aligned} x &= \hat{x} \cdot p \cdot \cos(\pi/6), \\ y &= \hat{\alpha} \cdot p + \hat{x} \cdot p \cdot \sin(\pi/6), \end{aligned}$$

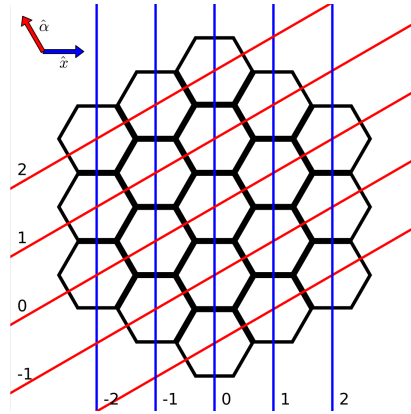
onde p é o *pitch* do reticulado. Para um reticulado orientado em ‘x’, basta trocar $\hat{\alpha}$ e \hat{x} na equação. Assim, para o núcleo do SPARK-NC, pode-se encontrar as coordenadas (x, y) do centro de um universo vareta/LBE circundante inserido em uma região hexagonal em um dado elemento, com tal centro de coordenadas tendo origem no centro do núcleo (deve ser lembrado que o nível de varetas é um reticulado orientado em ‘x’, enquanto o nível de elementos é orientado em ‘y’):

$$\begin{aligned} x_v &= \hat{x}_E \cdot p_E \cdot \cos(\pi/6) + \hat{x}_v \cdot p_v + \hat{\alpha}_v \cdot p_v \cdot \sin(\pi/6), \\ y_v &= \hat{\alpha}_E \cdot p_E + \hat{x}_E \cdot p_E \cdot \sin(\pi/6) + \hat{\alpha}_v \cdot p_v \cdot \cos(\pi/6), \end{aligned}$$

onde $(\hat{\alpha}_E, \hat{x}_E)$, p_E correspondem às coordenadas e *pitch* dos elementos no núcleo e $(\hat{\alpha}_v, \hat{x}_v)$, p_v são as coordenadas e *pitch* das varetas no elemento.

Figura 3.10 – Indexação $(\hat{\alpha}, \hat{x})$ para o reticulado hexagonal de 19 elementos. Retirado da documentação em

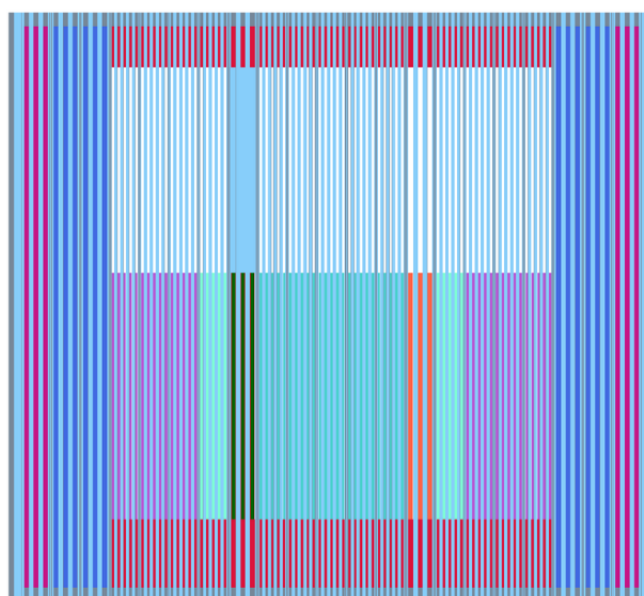
www.openmc.org



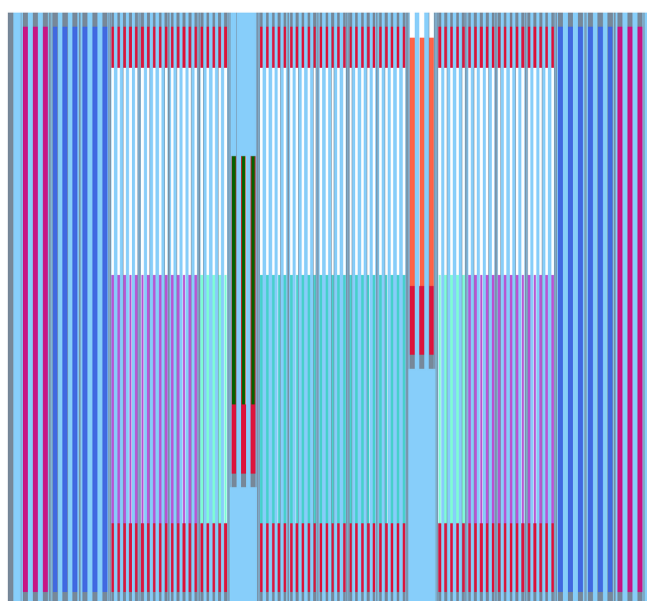
Quando o reticulado é ‘preenchido’ por um universo, todas as células desse universo são replicadas em diversas instâncias, sendo cada uma dessas instâncias inseridas em cada região desse reticulado. A numeração dessas instâncias obedecem um sequenciamento no OpenMC que permite o seu acesso de maneira ordenada (por exemplo, para se associar temperaturas às células de uma dada instância), e tal ordem é dada, em qualquer nível de reticulado, da seguinte maneira: Começa-se do valor menor possível para $\hat{\alpha}$ e \hat{x} e incrementa-se \hat{x} de 1 até atingir o valor máximo para aquele $\hat{\alpha}$, quando então incrementa-se $\hat{\alpha}$ de 1 e toma-se \hat{x} como o menor valor para esse novo $\hat{\alpha}$, e assim sucessivamente, até atingir o maior valor de $\hat{\alpha}$ e \hat{x} , no reticulado, que corresponde ao último valor de instância do universo no reticulado (na Figura 3.10, seria o equivalente a avançar de $\hat{\alpha} = -2, \hat{x} = 0$ até $\hat{\alpha} = -2, \hat{x} = 2$, então de $\hat{\alpha} = -1, \hat{x} = -1$ até $\hat{\alpha} = -1, \hat{x} = 2$ e assim sucessivamente).

A Figura 3.11 e a Figura 3.12 apresentam o resultado final da criação do objeto da classe **ReactorCore**, que é o universo representando toda a geometria de todo o núcleo com seus materiais. A Figura 3.12 mostra um corte transversal do núcleo, enquanto a Figura 3.11 mostra um corte longitudinal do núcleo num plano que passa pelo centro do mesmo, com as barras de controle totalmente inseridas e com um elemento de controle primário inserido em 60% no núcleo e um elemento de SCRAM inserido a 80% no núcleo. Tais figuras devem ser comparadas com suas contrapartes mostradas na Figura 3.3, que foram obtidas do artigo original do SPARK-NC [1].

Figura 3.11 – Corte longitudinal do núcleo do SPARK-NC modelado no OpenMC, com barras de controle/SCRAM totalmente inseridas (a) e com um elemento de controle primário inserido a 60% e um elemento de SCRAM inserido a 80% no núcleo (b).

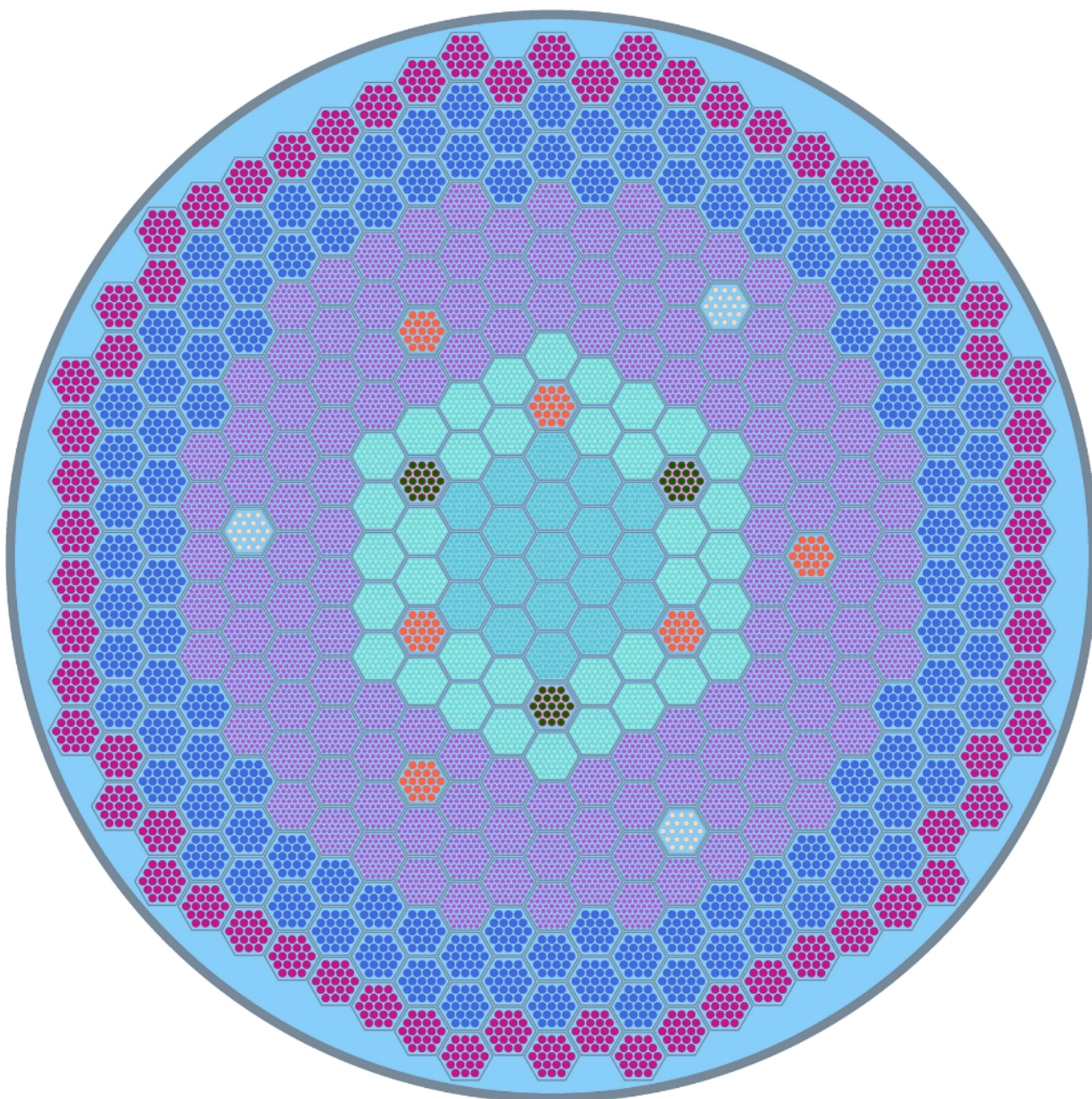


(a)



(b)

Figura 3.12 – Corte transversal do núcleo do SPARK-NC modelado no OpenMC.



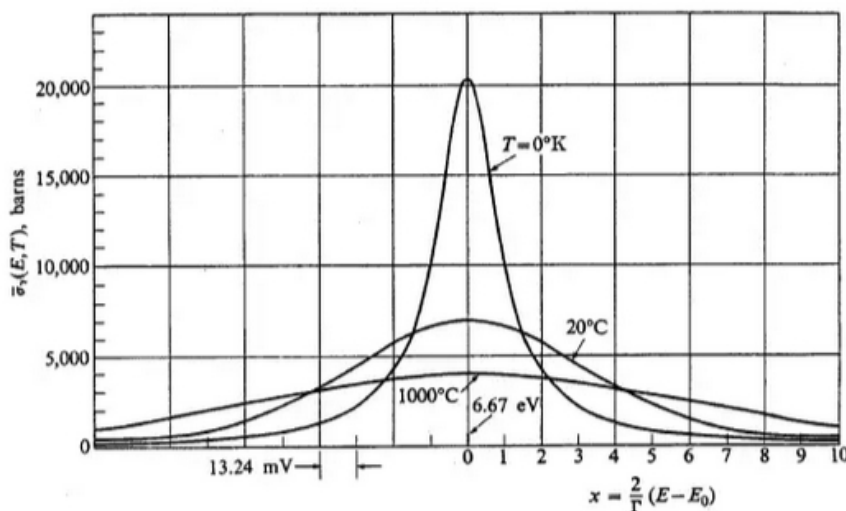
3.2 – O estágio de ajuste termohidráulico

3.2.1 – A descrição do processo de ajuste termohidráulico

Esta subseção irá detalhar o estágio de ajuste termohidráulico mencionado na Etapa 2 que foi discutida na introdução deste capítulo. Tal estágio é responsável por realizar um cálculo de criticalidade para o SPARK-NC, enquanto ajusta iterativamente as temperaturas/densidades das células do núcleo de acordo com a potência calorífica gerada no mesmo até se alcançar a convergência, obtendo-se então a distribuição de temperatura no núcleo correspondente ao estado estacionário de fluxo neutrônico associado com o valor encontrado de k_{eff} .

A distribuição de temperatura é um parâmetro crítico pois tem impacto direto sobre a neutrônica do núcleo; O efeito mais palpável desta relação é o *alargamento Doppler* das ressonâncias das seções de choque de captura de nêutrons pelos nuclídeos (ver Figura 3.13). A probabilidade de captura radiativa de nêutrons por um núcleo depende da velocidade relativa entre eles. Os núcleos dos átomos estão em movimento contínuo com velocidades de magnitude e direção variáveis devido às suas energias térmicas, e portanto mesmo um feixe de nêutrons monocromáticos impingindo em um núcleo-alvo é percebido por este núcleo em seu referencial de repouso como tendo uma faixa contínua de velocidades e, logo, de energias cinéticas. Assim, uma energia de nêutron incidente localizada numa região relativamente distante dos picos de ressonância de captura, caso o núcleo estivesse fixo, pode ser compensada pela velocidade de movimento do núcleo de modo que a velocidade relativa dos mesmos seja deslocada para a uma região de ressonância, favorecendo a captura do mesmo.

Figura 3.13 – Alargamento Doppler da ressonância na seção de choque microscópica de captura radiativa de nêutrons ($\bar{\sigma}_\gamma(E, T)$), com curvas de valores em barns para $T = 0\text{ K}$, 20°C e 100°C . Retirado de [7].



O aumento da temperatura do material aumenta a velocidade média de movimentação dos núcleos na estrutura cristalina do material, pois é proporcional a kT , sendo k a constante de Boltzmann e T a temperatura. Assim, a faixa de energia dos nêutrons que são capturados de maneira ressonante é efetivamente alargada em relação a um material “frio”, onde somente nêutrons com energias muito próximas do pico de ressonância podem ser absorvidos, conforme pode ser observado na Figura 3.13. Dado que os processos de captura diminuem a população de nêutrons disponíveis para fissão, o alargamento Doppler insere reatividade negativa no núcleo, tendo efeito redutor sobre sua criticalidade (de fato, o coeficiente de reatividade negativo de temperatura é altamente desejado como um fator de segurança intrínseco na operação do reator). Deste modo, um cálculo preciso de k_{eff} para um dado núcleo deve levar em consideração efeitos térmicos.

A densidade dos materiais também é um fator importante na neutrônica; por exemplo, o aumento da densidade de um dado material diminui o caminho livre médio dos nêutrons, favorecendo as interações e levando a uma distribuição de fluxo menos uniforme na região do material – o oposto ocorre quando diminui-se a densidade. Conforme mencionado na descrição da Etapa 2 na introdução deste capítulo, apenas o LBE é assumido como tendo uma variação de densidade significativa, por estar em estado líquido durante a operação e por ter uma interface de superfície livre com o restante do reator, representado pelos planos perpendiculares ao eixo z que definem a parte inferior e superior do núcleo na geometria, o que permite a sua expansão.

A proposta do autor para este estágio é realizar um cálculo de criticalidade para o núcleo no OpenMC, ao mesmo tempo em que é feito um ajuste adaptativo da distribuição de temperatura de todo o núcleo e das densidades das células preenchidas por LBE de acordo com esta distribuição. O OpenMC oferece uma biblioteca de funções que permite realizar uma simulação gradual e adaptativa de criticalidade: É possível interromper o cálculo de criticalidade após um determinado número de bateladas e extrair dados do estado do núcleo no instante imediatamente anterior à interrupção, como valores de *tallies* e de parâmetros dos materiais de das células deste núcleo, ao mesmo tempo em que se pode modificar tais parâmetros, retomando o cálculo de criticalidade com o núcleo atualizado. Tal ciclo de interrupção/atualização/retomada pode ser realizada quantas vezes e em qualquer ponto do cálculo em que se desejar, até que se atinja o número total de bateladas especificadas para a obtenção do valor final de k_{eff} .

As funções supramencionadas são propícias ao ajuste adaptativo proposto, pois permite que os *tallies* de potência calorífica gerada em cada uma das varetas combustíveis do núcleo do SPARK-NC sejam obtidos em intervalos regulares durante o cálculo de criticalidade, enviando

então tais dados para um módulo termohidráulico que irá fornecer a distribuição de temperatura das células e densidades dos materiais LBE, sendo finalmente tais distribuições atribuídas às respectivas células e materiais do núcleo antes da retomada do cálculo de criticalidade, até um novo ciclo de interrupção/atualização/retomada ou a finalização do cálculo. Deste modo, o ajuste torna-se uma espécie de método de solução iterativa, onde espera-se que tanto a distribuição de temperaturas/densidades quanto o valor de k_{eff} convirjam após um determinado número de ciclos.

A convergência da solução iterativa é um indicativo de que obteve-se uma solução de estado estacionário de fluxo neutrônico no núcleo correspondente ao valor encontrado de k_{eff} (ou seja, a solução do problema de autovalores para a geometria e os materiais definidos para o núcleo), e também indica que a distribuição de temperaturas encontrada é também uma distribuição de estado estacionário induzida por aquele estado de estacionário de fluxo. Logo, espera-se que o valor de k_{eff} obtido ao final do ajuste termohidráulico é aquele que melhor corresponde à condição térmica do reator, dadas as aproximações numéricas e os desvios estatísticos inerentes aos cálculos, sendo portanto o valor utilizado para fins de comparação com a curva da Figura 3.1 (ver fluxograma na Figura 3.2).

Os itens seguintes mostrarão a descrição do processo de ajuste, enquanto a Subseção 3.2.2 dedicar-se-á à modelagem do processo no OpenMC.

3.2.1.1 – O cálculo da distribuição de temperatura nos elementos combustíveis

Primeiramente, mostra-se o processo que leva ao cálculo distribuição de temperatura nas varetas nos elementos combustíveis, que são os elementos onde a potência calorífica do reator é produzida. O primeiro passo é obter a distribuição da densidade de potência nas varetas dos elementos. O OpenMC não fornece tal grandeza diretamente, mas oferece um *tally* (ver a definição de *tally* no Capítulo 2) que entrega o calor produzido em uma região ζ do núcleo:

$$Q = \int_{\zeta} d^3r \int_0^{\infty} dE h(E, \vec{r}) \oint_{U_1} d\Omega \psi(\vec{r}, \hat{\Omega}, E) = \int_{\zeta} d^3r \int_0^{\infty} dE h(E, \vec{r}) \phi(\vec{r}, E)$$

onde $h(E, \vec{r})$ é uma função angularmente isotrópica (isto é, independe da direção de incidência do nêutron $\hat{\Omega}$) de deposição de energia calorífica no material, que depende da composição do mesmo e da energia incidente dos nêutrons sobre os núclídeos que o constituem, e $\psi(\vec{r}, \Omega, E)$ e $\phi(\vec{r}, E)$ são, respectivamente, o fluxo de nêutrons e o fluxo de nêutrons integrado no ângulo. Como qualquer *tally* obtido em Monte Carlo, os valores são normalizados pelo número de partículas-fonte do

problema (ver Seção 2.4), e portanto, as unidades de Q são elétron-Volt/partícula-fonte. De modo a converter esta grandeza para a densidade volumétrica de potência \dot{q} em unidades mais palatáveis para a inserção em um código para o cálculo de temperatura nas varetas, como W/cm^3 , necessita-se multiplicar tais valores por um fator de conversão. Tal fator pode ser calculado da seguinte maneira [8]:

1. Faz-se um *tally* separado de produção calorífica onde o volume V abrange todo o núcleo do reator, gerando o valor Q_R ;
2. Toma-se a potência térmica total do reator e divide-se por Q_R , resultando em um fator com unidade Joule.partículas-fonte por elétron-Volt.segundo:

$$f = \frac{P}{Q_R} = \frac{[J/s]}{[eV/pf]} = \left[\frac{J.pf}{eV.s} \right]$$

3. O fator f pode ser agora utilizado para se converter qualquer *tally* de unidade eV /partícula-fonte para J/s . Portanto, se Q é o *tally* de produção calorífica em uma dada região ζ do núcleo em $[eV/pf]$ e V é o volume dessa região em cm^3 , o *tally* nessa região é convertido para a densidade de potência \dot{q} em (W/cm^3) fazendo -se:

$$\dot{q} = \frac{f \cdot Q}{V} \left[\frac{J.pf}{eV.s} \right] \left[\frac{eV}{pf} \right] \left[\frac{1}{\text{cm}^3} \right] = \left[\frac{J}{s.\text{cm}^3} \right] = \left[\frac{W}{\text{cm}^3} \right].$$

Em cada vez que o cálculo de criticalidade é interrompido para um ciclo de atualização de temperaturas/densidades, o valor de um *tally* de produção calorífica de uma dada região não é necessariamente igual ao do valor para esta mesma região obtida em ciclos diferentes, pois o fluxo $\psi(\vec{r}, \Omega, E)$ ainda está em processo de ajuste durante o cálculo de criticalidade e portanto variará entre um ciclo e outro (principalmente entre os ciclos iniciais). Isto significa que o fator de conversão f mostrado acima também não é constante entre ciclos, e portanto o *tally* Q de uma dada região extraído durante um ciclo deve ser normalizado por um fator f que é calculado fazendo os passos dados acima, a partir de um *tally* que abrange todo o reator que é também extraído durante esse ciclo.

Obtido o valor de \dot{q} para uma dada região em um dado ciclo em unidades “naturais” (W/cm^3 , W/mm^3 , etc.), pode-se calcular a distribuição de temperatura nesta região, se a mesma for preenchida por um mesmo material, ao se solucionar a equação do calor não-homogênea:

$$\rho C_p \frac{\partial T(\vec{r}, t)}{\partial t} - \lambda \nabla^2 T(\vec{r}, t) = \dot{q}(\vec{r}, t),$$

sendo $T(\vec{r}, t)$ e $\dot{q}(\vec{r}, t)$, respectivamente, a temperatura e a densidade volumétrica de potência no instante t e no ponto \vec{r} da região, e λ, C_p, ρ respectivamente, os valores da condutividade térmica, calor específico à pressão constante e densidade para o material que preenche a região. Visto que se deseja apenas a temperatura em estado estacionário ($\frac{\partial}{\partial t} = 0$), que os elementos do reator possuem diversas regiões preenchidas por materiais distintos, e as propriedades térmicas de um dado material podem depender da própria temperatura do mesmo, a equação acima deve ser modificada para contemplar tais considerações:

$$-\nabla \cdot [\lambda(\vec{r}, T) \nabla T(\vec{r})] = \dot{q}(\vec{r})$$

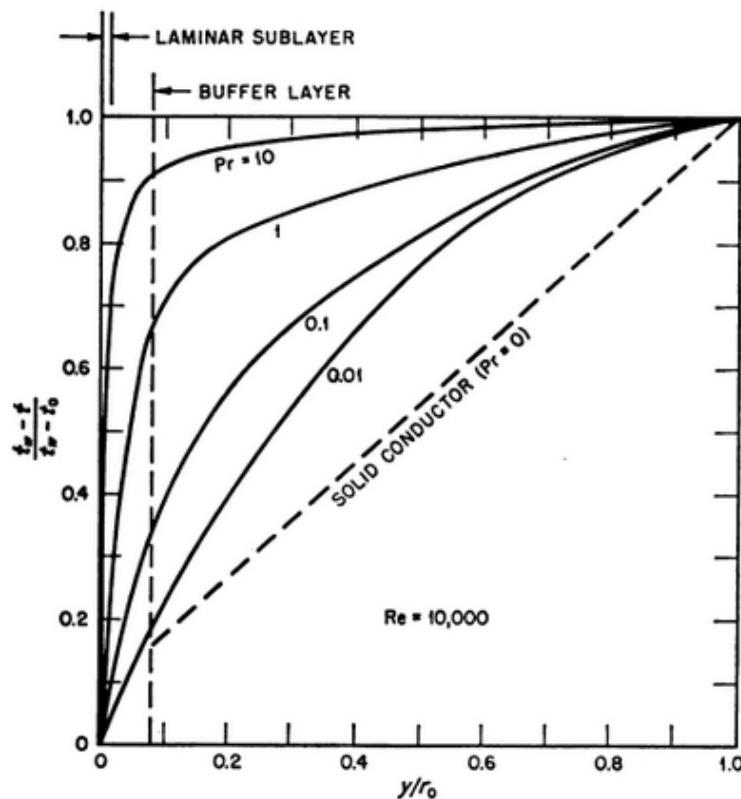
onde agora $\lambda(\vec{r}, T)$ corresponde à condutividade térmica que depende do ponto \vec{r} e da temperatura nesse ponto. Porém, sabe-se que as varetas e o elemento são circundados pelo fluido refrigerante, que circula para a dissipação de calor no núcleo e portanto não permanece fixo no reator como os outros materiais. Uma solução plausível para a distribuições de temperatura no elemento deve levar tal fato em consideração, e a análise portanto deve ser não somente térmica, mas termohidráulica.

A modelagem termohidráulica do núcleo de reatores é um assunto abrangente, possuindo diversos graus de complexidade de acordo com as especificidades do reator e do grau de minúcia da abordagem; em fluidos refrigerantes ordinários, como a água leve, o principal mecanismo de transporte de calor é o efeito de turbulência, com a massa de fluido turbulento deslocando-se rapidamente das regiões próximas da camada laminar de fluido que adere à parede do revestimento combustível até a região principal de corrente do fluido [9]. A modelagem de tal fenômeno é complexa sob os aspectos teóricos e computacionais, onde frequentemente assume-se hipóteses específicas ao problema em questão. No entanto, em metais líquidos como o LBE, o transporte de calor se dá principalmente por condução térmica, correspondendo a cerca de 70% da transmissão [9] (para fins de comparação, na água leve a condução é responsável por apenas 0,2% da transmissão de calor). Isto significa que a espessura da camada limite laminar do fluido próxima à parede do revestimento não tem importância significativa para metais líquidos [9].

A diferença essencial entre as propriedades de transmissão de calor dos metais líquidos e dos fluidos ordinários pode ser visto na Figura 3.14, retirada de [9]. O gráfico mostra os perfis de temperatura de diversos fluidos com número de Reynolds (Re) = 10000 que atravessam o interior de um tubo aquecido, de acordo com o número de Prandtl (Pr) do fluido. A abcissa mostra a razão entre a distância do fluido para a parede do tubo (y) e o raio do tubo (r_0). A ordenada mostra a razão $\frac{t_w - t}{t_w - t_0}$, sendo t_w a temperatura da parede, t_0 a temperatura do centro do tubo e t a temperatura do

ponto y . Deste modo, tal razão denota quão “frio”, numa escala de 0 a 1, o ponto y está em relação à parede do tubo. Pode-se ver que para fluidos ordinários como a água leve ($10 < Pr < 1$) a temperatura cai bruscamente já na região da sub-camada limite laminar (*laminar sublayer*), já tendo grande parte da transferência ocorrida na região quase laminar da camada intermediária (*buffer layer*). Já para metais líquidos como o LBE, que possui um número de Prandtl significativamente menor ($0.1 < Pr < 0.01$), o comportamento é similar àquele puramente condutivo de um sólido ($Pr = 0$): a distribuição de temperatura é quase linear em direção centro do tubo, indicando que o deslocamento do fluido tem um efeito bem menor sobre a transmissão do calor, pois para os metais líquidos a velocidade de condução térmica molecular é dominante sobre a velocidade de transporte de massa para $Re = 10000$. Também nota-se que, para números de Prandtl baixos, não há diferença marcante na inclinação da curva quando as regiões laminar e intermediária são comparadas com o restante do tubo e portanto a região laminar não oferece distinção significativa, em termos de transmissão de calor, para metais líquidos, conforme mencionado.

Figura 3.14 – Curva de condução térmica de fluidos no interior de um tubo de raio r_0 com paredes aquecidas à temperatura t_w , para diversos números de Prandtl, com número de Reynolds = 10000. *Laminar sublayer* corresponde à subcamada limite laminar, enquanto *buffer layer* corresponde à camada intermediária que contém a transição entre os regimes laminar e turbulento. Retirado de [9].



Conforme pode ser visto na Tabela 3.4, a velocidade média especificada para o fluido refrigerante no SPARK-NC é de 0.31 m/s. Tomando valores médios de propriedades físicas do LBE

dentro da temperatura de operação do reator [6], e considerando o diâmetro hidráulico para um reticulado hexagonal como a dimensão característica, pode-se ter uma aproximação para o número de Reynolds médio do LBE em um elemento combustível do SPARK-NC:

$$L = d \left[\frac{2\sqrt{3}}{\pi} \left(\frac{p}{d} \right)^2 - 1 \right] = 0.0069 \text{ m},$$

$$Re = \frac{\rho |\vec{v}| L}{\mu} = \frac{(1.012 \times 10^4 \frac{\text{kg}}{\text{m}^3})(0.31 \frac{\text{m}}{\text{s}})(0.0069 \text{ m})}{1.376 \times 10^{-3} \frac{\text{kg}\cdot\text{s}}{\text{m}}} \simeq 15700.$$

Na fórmula, d é o diâmetro de uma vareta combustível (1 cm), p é o *pitch* do elemento (1.24 cm), ρ e μ são, respectivamente, a densidade e a viscosidade dinâmica do LBE, \vec{v} é a velocidade média especificada para o LBE e L é a dimensão característica. Este número é próximo e da mesma ordem de grandeza de $Re = 10000$, e portanto espera-se que o LBE no SPARK-NC tenha o mesmo comportamento, em termos de transferência de calor, da Figura 3.14. Apesar da transferência de calor em metais líquidos ser eminentemente condutiva, deve-se atrelar ao modelo a informação de que o calor é removido pela passagem do fluido refrigerante. Deste modo, o autor optou por uma abordagem relativamente simples, incorporando um termo convectivo ao modelo térmico com a velocidade média \vec{v} do fluido refrigerante especificada para o SPARK-NC (ver Tabela 3.4):

$$-\nabla \cdot (\lambda \nabla T) + \nabla \cdot (\rho C_p \vec{v} T) = \dot{q}$$

onde $\dot{q}, \lambda, C_p, \rho \equiv \dot{q}(\vec{r}), \lambda(\vec{r}, T), C_p(\vec{r}, T), \rho(\vec{r}, T)$ são, respectivamente, a densidade volumétrica de potência no ponto \vec{r} e a condutividade térmica, o calor específico à pressão constante e a densidade no ponto \vec{r} e na temperatura T deste ponto. Visto que \vec{v} é não-nula apenas nas células contendo fluido refrigerante, pela forma da equação nota-se que é necessário especificar estas duas últimas grandezas apenas para o LBE. Deve ser mencionado que o epíteto “termohidráulico” empregado aqui deve-se tão somente ao fato de que o modelo leva em consideração a velocidade média do fluido refrigerante nos canais das varetas do núcleo para representar a remoção do calor através de convecção; um modelo termohidráulico em estado estacionário mais completo teria $\vec{v} \equiv \vec{v}(\vec{r}, T)$, visto que a velocidade seria variante no espaço e dependeria da temperatura, sendo descrita pelas equações de dinâmica de fluidos envolvendo grandezas também termo-dependentes como a pressão, a densidade e a viscosidade, através das relações de conservação de massa e *momentum* (equações de Navier-Stokes). Apesar da construção de tal modelo ser factível (e indispensável para a acurácia de alguns modelos envolvendo fluidos ordinários), pode-se

argumentar, com base na discussão sobre o comportamento térmico de metais líquidos mostrada nos parágrafos acima, que o fato de se empregar uma velocidade média \vec{v} no domínio espacial correspondente ao LBE não possui um impacto significativo sobre o modelo, visto que, devido ao fato de que a condução térmica domina sobre o transporte de massa nestes fluidos para números de Reynolds relativamente baixos como os considerados em questão, o termo difusivo, relativo à condução térmica molecular (isto é, o termo $-\nabla \cdot (\lambda \nabla T)$) é dominante sobre o termo convectivo (o termo $\nabla \cdot (\rho C_p \vec{v} T)$), e portanto os resultados do modelo serão relativamente insensíveis no que se refere ao emprego da velocidade média \vec{v} em detrimento do mapa preciso de velocidades $\vec{v}(\vec{r}, T)$ ao longo do domínio, o que permite modelar efeitos dinâmicos como turbulência por meio da modificação da condutividade térmica molecular λ por um valor efetivo λ_{eff} que leve em consideração tais efeitos através de relações semi-empíricas. Não obstante, sugere-se, como um direcionamento futuro, incorporar as equações de Navier-Stokes ao modelo OpenMC/FiPy para um reator refrigerado por um fluido arbitrário cujas curvas térmicas e condições de contorno são conhecidas, de modo a comparar as curvas e parâmetros de desempenho associados (ver Seção 5.4).

O modelo mostrado acima pode ser aplicado a todo o núcleo, calculando-se com o emprego da equação de difusão-convecção a distribuição de temperatura em todas as células através das propriedades dos materiais e das distribuições espaciais de densidade de potência. Porém, tal empreendimento é computacionalmente dispendioso, dado o número de células que compõem a geometria do núcleo, e o fato que a região correspondente à cada célula deve ser, em geral, subdividida em uma malha para a solução numérica das equações acima. Assim, a distribuição de temperatura em todo o núcleo deve ser calculada de maneira indireta através da distribuição de temperatura em apenas uma região do mesmo, utilizando-se de considerações fenomenológicas para mapear a temperatura de tal região para o restante do núcleo.

O autor emprega uma abordagem similar àquela discutida em [10], em que se calcula, em cada ciclo de interrupção/ajuste/atualização, a distribuição de temperatura de apenas uma região vareta+LBE circundante representativa, que no caso é a região correspondente à vareta combustível que gerou a maior potência calorífica naquele ciclo. Visto que todas as regiões das varetas combustíveis/LBE circundante são subdivididas da mesma forma e pelo mesmo número de células, a temperatura de uma célula c de uma região correspondente à uma vareta combustível qualquer v tem sua temperatura T_c^v atribuída através da temperatura T_c^R , da célula correspondente na região da vareta representativa R , pela seguinte relação:

$$T_c^v = T_b + \alpha_v(T_c^R - T_b), \quad \alpha_v \equiv \frac{Q_v}{Q_R}.$$

Nesta equação, α_v é a razão entre as produções caloríficas geradas na vareta v considerada e na vareta representativa, e $T_b = 330\text{ }^{\circ}\text{C} \approx 600\text{ K}$, que é a temperatura de entrada do LBE no núcleo. Tal temperatura é empregada como temperatura de base nessa relação porque, se não houvesse aquecimento dos componentes do núcleo pela potência gerada pelas varetas, tais componentes estariam aproximadamente à temperatura T_b após entrarem em equilíbrio térmico com o LBE circulante; deste modo, espera-se que T_b seja a menor temperatura encontrada no núcleo em estado estacionário. A temperatura T_c^R da célula da vareta representativa é obtida ao se resolver a equação de convecção-condução para a região correspondente à vareta representativa com condições de contorno apropriadas, e tendo-se como fonte da equação a distribuição $\dot{q}(\vec{r}) = \dot{q}_R(\vec{r})$, isto é, a distribuição de densidade volumétrica de potência gerada na vareta representativa, atribuindo-se então como a temperatura da célula c da vareta representativa a média da distribuição de temperatura calculada $T^R(\vec{r})$ nos pontos \vec{r} pertencentes à célula.

O mapa linear dado pela equação é uma boa aproximação se as produções caloríficas de todas as varetas combustíveis são próximas às da vareta representativa (isto é, se α_v não é muito próximo de zero ou muito maior que um); se isto não ocorre, a não-linearidade advinda da dependência dos parâmetros térmicos com a temperatura faz com que a diferença de temperaturas entre T^v e T^R se distancie significativamente de uma relação diretamente proporcional à produção calorífica relativa. No entanto, a proximidade de α_v com a unidade é esperada, visto que a região ativa do SPARK-NC possui propriedades homogêneas, é simulado em uma condição em que suas barras de controle/SCRAM estão suspensas e seu núcleo é projetado para que a distribuição radial de potência seja suave [1].

Observa-se que a equação mantém o perfil de temperatura encontrado na região da vareta representativa, ajustando apenas a sua escala de acordo com a produção calorífica relativa α_v , que é sempre menor ou igual a 1. Também, nota-se que para a vareta representativa a produção calorífica relativa é 1 e $T_c^v = T_c^R$, como esperado. Após a obtenção das temperaturas das células correspondentes às varetas/LBE circundante, a temperatura das células que compõem o revestimento de cada elemento é tomada como a média das temperaturas de todas as células de LBE circundante às varetas que estão naquele elemento; faz-se isso pois é o LBE que permanece em contato com as paredes do revestimento no interior do elemento (ver Figura 3.8).

De acordo com a discussão dos parágrafos acima, para obter-se T_c^v , deve-se extrair $\dot{q}_R(\vec{r})$, isto é, a distribuição de densidade volumétrica de potência na região da vareta que gerou a maior potência no núcleo em um dado ciclo, de modo a calcular os valores de T_c^R através da equação de difusão-convecção, e o fator α_v que mapeia entre T_c^R e T_c^v para cada vareta v . Neste trabalho, utiliza-se uma abordagem que extrai os valores de $\dot{q}_R(\vec{r})$ e α_v através de *tallies* de produção

calorífica sobre a região do cerne de cada uma das varetas combustíveis, além do *tally* que fornece a densidade linear (isto é, por unidade de comprimento), na direção axial, da produção calorífica na região ativa do núcleo.

O OpenMC oferece *tallies* de densidade axial linear (isto é, por unidade de comprimento) de produção calorífica $Q(z)$ e de produção calorífica radial-azimutal por unidade de área $Q(r, \theta)$ em uma região cilíndrica do núcleo $\zeta : \{0 \leq r \leq R, 0 \leq \theta \leq 2\pi, z_{min} \leq z \leq z_{max}\}$, aproximando-as pelas seguintes expansões funcionais:

$$Q(z) \approx \sum_{n=0}^{N_z} a_n P_n(f(z)) \quad Q(r, \theta) \approx \sum_{p=0}^{N_r} \sum_{q=-p}^p b_p^q Z_p^q\left(\frac{r}{R}, \theta\right),$$

onde $f(z)$ é uma função que mapeia de (z_{min}, z_{max}) para $(-1, 1)$,

$$P_n(z) = \frac{1}{2^n n!} \frac{d^n}{dz^n} [(z^2 - 1)^n]$$

são os **polinômios de Legendre**, que são definidos no domínio $z \in (-1, 1)$ e

$$\begin{aligned} Z_p^q(r, \theta) &= R_p^q(r) \cos q\theta \\ Z_p^{-q}(r, \theta) &= R_p^q(r) \sin q\theta \end{aligned} \quad R_p^q(r) = \frac{\Gamma(p+1) {}_2F_1(-\frac{1}{2}(p+q), \frac{1}{2}(q-p); -p; r^{-2}) r^p}{\Gamma(\frac{1}{2}(2+p-q)) \Gamma(\frac{1}{2}(2+p+q))},$$

são os **polinômios de Zernike** definidos no círculo unitário $(0, 1) \otimes (0, 2\pi)$, com $p \geq q \geq 0$, onde $\Gamma(x)$ é a função Gama e ${}_2F_1(a, b; c; z)$ a função Hipergeométrica. Nas equações acima, nota-se que as expansões de $Q(z)$ e $Q(r, \theta)$ são, respectivamente, de ordem N_z e N_r , com as aproximações tornando-se exatas quando a ordem de ambos é infinita. Deve ser observado que, dado que $Q(z)$ é a densidade linear axial de produção calorífica e Q a produção calorífica total na região ζ , então

$$\int_{z_{min}}^{z_{max}} Q(z) dz = Q \equiv \int_{\zeta} d\vec{r} \int_0^\infty dE h(\vec{r}, E) \phi(\vec{r}, E) = \int_{z_{min}}^{z_{max}} dz \int_0^{2\pi} d\theta \int_0^R r dr \int_0^\infty dE h(E, r, \theta, z) \phi(E, r, \theta, z)$$

com $\phi(\vec{r}, E) \equiv \oint_{U_1} d\hat{\Omega} \psi(\vec{r}, \hat{\Omega}, E)$ o fluxo neutrônico escalar e $h(\vec{r}, E)$ a função angularmente isotrópica de deposição calorífica. Logo

$$Q(z) = \int_0^{2\pi} d\theta \int_0^R r dr \int_0^\infty dE h(E, r, \theta, z) \phi(E, r, \theta, z) \equiv \int_0^{2\pi} d\theta \int_0^R r dr q(r, \theta, z),$$

sendo $q(r, \theta, z) \equiv \int_0^\infty dE h(E, r, \theta, z) \phi(E, r, \theta, z)$ a densidade volumétrica de produção calorífica

no núcleo em coordenadas cilíndricas. De maneira semelhante, conclui-se que

$$Q(r, \theta) = \int_0^\infty dE \int_{z_{min}}^{z_{max}} dz h(r, \theta, z, E) \phi(r, \theta, z, E).$$

Devido à ortogonalidade dos polinômios de Legendre, pode ser mostrado que, após transformar os limites $z \in (z_{min}, z_{max})$ do domínio cilíndrico para $z' \in (-1, 1)$, através de uma função $z' = f(z)$, os coeficientes de $P_n(z)$ são dados por:

$$\begin{aligned} a_n &= \frac{(2n+1)}{2} \int_{-1}^1 dz' P_n(z') Q(f^{-1}(z')) = \\ &= \int_{z_{min}}^{z_{max}} dz \int_0^\infty dE \int_0^{2\pi} d\theta \int_0^R r dr \left[\frac{(2n+1)}{2} \frac{df(z)}{dz} P_n(f(z)) h(r, \theta, z, E) \right] \phi(r, \theta, z, E) \\ &\equiv \int_{\zeta} d^3r \int_0^\infty dE g_n(\vec{r}, E) \phi(\vec{r}, E) = \int_{\zeta} d^3r \int_0^\infty dE \oint_{U_1} d\hat{\Omega} g_n(\vec{r}, E) \psi(\vec{r}, \hat{\Omega}, E), \end{aligned}$$

tomando-se $\vec{r} = (r, \theta, z)$, $d^3r = r dr dz d\theta$, $\zeta = (z_{max}, z_{min}) \otimes (0, R) \otimes (0, 2\pi)$, e a função entre colchetes igual a $g_n(r, \theta, z, E) \equiv g_n(\vec{r}, E)$. Portanto, a obtenção do coeficiente a_n resume-se ao cálculo do *tally* na região ζ com a função resposta (*score*) dada por

$$g_n(\vec{r}, E) \equiv g_n(r, \theta, z, E) = \frac{(2n+1)}{2} \frac{df(z)}{dz} P_n(f(z)) h(r, \theta, z, E).$$

Da mesma forma, tem-se para os coeficientes b_q^p da expansão em polinômios de Zernike, com $r \in (0, R)$ normalizado para $r' \in (0, 1)$ através de $r' = r/R$, $dr' = dr/R$:

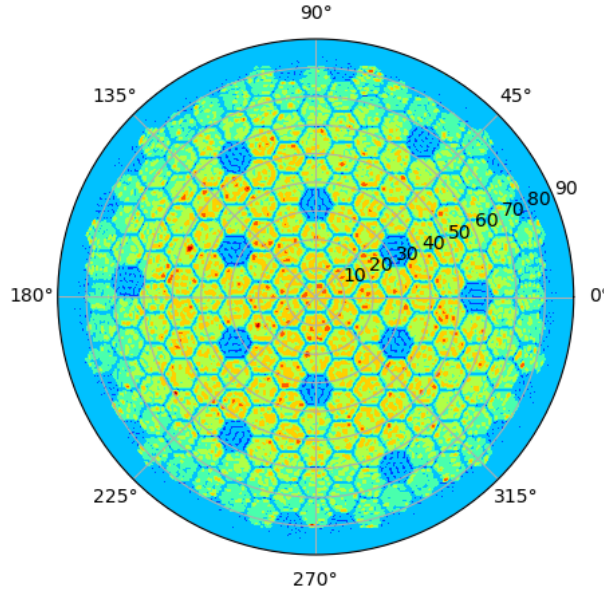
$$b_p^q = k_p^q \int_0^1 r' dr' \int_0^{2\pi} d\theta Z_p^q(r', \theta) Q(r', \theta) \quad k_p^q = \begin{cases} \frac{2p+2}{\pi}, & q \neq 0 \\ \frac{p+1}{\pi}, & q = 0 \end{cases}.$$

Pode ser então mostrado que b_p^q é dado por

$$b_p^q = \int_{\zeta} d^3r \int_0^{\infty} dE g_p^q(\vec{r}, E) \phi(\vec{r}, E), \quad g_p^q(\vec{r}, E) \equiv g_p^q(r, \theta, z, E) = \frac{k_p^q}{R^2} Z_p^q\left(\frac{r}{R}, \theta\right) h(r, \theta, z, E),$$

ou seja, o problema do cálculo dos coeficientes também reduz-se a *tallies* com o *score* g_p^q na região ζ . Uma das vantagens de tais expansões funcionais é que, mesmo truncando as séries em ordens relativamente baixas (e portanto realizando o cálculo de um número reduzido de *tallies*), é possível obter uma aproximação contínua no espaço para as distribuições de grandezas físicas na região ζ , com um nível de precisão que só seria possível com uma malha composta por um número muito grande de *tallies* ordinários subdividindo a região em questão. Para fins de ilustração, a Figura 3.15 fornece $Q(r, \theta)$ na região ativa do núcleo do SPARK-NC com a expansão até a ordem $N_r = 300$.

Figura 3.15 - Densidade por unidade de área radial-azimutal de produção calorífica na região ativa do núcleo $Q(r, \theta)$, obtida por expansão funcional em polinômios de Zernike até a ordem 300.



Conforme já mencionado, $Q(z)$ é uma densidade linear de produção calorífica na direção axial: pela definição de $Q(z)$, nota-se que $\int_{z_{min}}^{z_{max}} Q(z) dz$ equivale ao *tally* Q que fornece o calor total gerado em toda a região ζ e que foi definido no primeiro parágrafo desta seção. Portanto, se ζ for tomada como a região cilíndrica em que o raio R compreende todas as varetas combustíveis no núcleo, e z_{min}, z_{max} são, respectivamente, a altura mínima e máxima do cerne das varetas, e for

assumido que o calor for gerado exclusivamente no cerne combustível das varetas nessa região, isto é, $q(r, \theta, z) = 0$ fora do volume que correspondente ao cerne das varetas, sendo $q(r, \theta, z) \equiv \int_0^\infty dE h(E, r, \theta, z) \phi(E, r, \theta, z)$ a já mencionada distribuição de densidade volumétrica de produção calorífica no núcleo, então $Q(z) = \sum_v \iint_{A_v} r dr d\theta q(r, \theta, z)$, onde A_v é a área transversal de uma dada vareta v e a soma \sum_v é feita sobre todas as varetas combustíveis v do núcleo. Portanto, ao se dividir $Q(z)$ pela área A que corresponde à soma das áreas de seção transversal do cerne de todas as varetas na região ζ que abrange todos os elementos combustíveis do núcleo, isto é, $A = \sum_v A_v$, tem-se:

$$\frac{Q(z)}{A} = \frac{\sum_v \iint_{A_v} r dr d\theta q(r, \theta, z)}{\sum_v A_v} = \frac{\sum_v q_v(z) \iint_{A_v} r dr d\theta}{\sum_v A_v} = \frac{\sum_v q_v(z) A_v}{\sum_v A_v} = \frac{\sum_v q_v(z)}{N},$$

sendo

$$q_v(z) = \left[\iint_{A_v} q(r, \theta, z) r dr d\theta \right] \cdot \left[\iint_{A_v} r dr d\theta \right]^{-1} = \frac{1}{A_v} \iint_{A_v} q(r, \theta, z) r dr d\theta$$

o valor médio de $q(r, \theta, z)$ na região $(r, \theta) \in A_v$ correspondente à seção de área transversal da vareta v e N é o número total de varetas combustíveis. Na última equação, assume-se que a área de seção transversal A_v é a mesma para todas as varetas combustíveis, o que é uma boa aproximação, visto que 90% das varetas possuem um diâmetro de 1 cm e o restante possui um diâmetro de 0.96 cm (ver Tabela 3.1), sendo também a diferença de áreas transversais entre esses dois tipos de varetas de apenas 7%. Nota-se que o último termo da equação acima dá a média de $q_v(z)$ tomada entre todas as varetas combustíveis, isto é, o perfil axial médio, dentre todas as varetas combustíveis, de densidade volumétrica de produção calorífica, doravante denominado $\bar{q}(z)$. Assim, $\bar{q}(z) = Q(z)/A$.

Assumindo-se também que o formato do perfil axial de densidade de potência volumétrica nas varetas combustíveis varia pouco em torno da média, pode-se considerar que $q_v(z)$, o perfil axial de densidade volumétrica de produção calorífica de uma vareta combustível v qualquer do núcleo, terá um formato aproximadamente igual ao perfil axial médio $\bar{q}(z)$ de densidade

volumétrica, tendo apenas sua magnitude alterada por um fator β_v que depende de cada vareta v . Assim, $q_v(z) = \beta_v \bar{q}(z)$.

Deve ser observado que as hipóteses feitas acima são plausíveis para o reator considerado: A produção calorífica é largamente devida às fissões que ocorrem no cerne das varetas combustíveis, com as outras formas de geração de calor, como a deposição nos materiais da energia dos fótons que são produtos de reações (n, γ) , fornecendo contribuições relativas residuais. Portanto, considera-se com boa aproximação que $q(r, \theta, z) = 0$ fora das varetas. Do mesmo modo, os cálculos são feitos com as barras de controle/SCRAM levantadas, ou seja, não há presença de materiais fortemente absorvedores na região ζ , e os materiais combustíveis estão distribuídos de maneira uniforme e tem propriedades aproximadamente uniformes na região. Deste modo, pode-se também assumir, com boa aproximação, que os perfis axiais de potência das varetas são similares e não terão diferenças significativas de formato em relação à média, apenas de magnitude devida à distribuição radial de intensidade de fluxo no núcleo. Na Seção 4.2 são mostrados dados que corroboram a plausibilidade de tais hipóteses.

A distribuição volumétrica de potência no cerne combustível da vareta representativa, $\dot{q}_R(\vec{r})$, necessária ao cálculo da distribuição de temperatura na mesma, é, na sua forma mais precisa, dada pelo produto entre o fator f que converte a produção calorífica em potência mencionado no início desta subseção, e a densidade volumétrica de produção calorífica na vareta representativa, isto é, $\dot{q}_R(\vec{r}) \equiv \dot{q}_R(r, \theta, z) = f q_R(r, \theta, z)$. Porém, visto que pode-se obter, conforme mencionado acima, $q_R(z) = \beta_R \bar{q}(z) = \beta_R Q(z)/A$, onde β_R é o fator de magnitude para a vareta representativa e $q_R(z)$ é o valor médio da densidade volumétrica de potência $q(r, \theta, z)$ na região da seção de área transversal correspondente à altura z da vareta representativa, então a simulação numérica de $T^R(\vec{r})$ será feita tendo como distribuição de densidade volumétrica de potência o valor $\dot{q}_R(z) = f q_R(z)$ em cada ponto da seção transversal correspondente à altura z do cerne combustível. Tal distribuição assume que a distribuição de potência não varia significativamente dentro da área de seção transversal de uma vareta; porém, tal aproximação é razoável ao se considerar que a distribuição radial de potência no SPARK-NC é suave, conforme já mencionado, devido ao seu projeto e ao fato que a simulação será feita sem a presença de absorvedores inseridos em sua região ativa.

Será mostrada uma fórmula explícita para o supramencionado fator β_v relativo a uma vareta combustível qualquer v do núcleo. Primeiramente, tem-se, conforme já discutido acima:

$$q_v(z) = \beta_v \bar{q}(z) = \beta_v \left[\frac{1}{N} \sum_p q_p(z) \right],$$

sendo o somatório sobre todas as varetas combustíveis e N o número total de varetas combustíveis. Em seguida, lembrando que $q_v(z) = \frac{1}{A_v} \iint_{A_v} q(r, \theta, z) r dr d\theta$, onde $q(r, \theta, z)$ é a distribuição de densidade volumétrica de produção calorífica no núcleo e A_v é a área de seção transversal da vareta v , tem-se

$$\int_{z_{min}}^{z_{max}} q_v(z) dz = \frac{1}{A_v} \iint_{A_v} \left[\int_{z_{min}}^{z_{max}} q(r, \theta, z) dz \right] r dr d\theta = \frac{Q_v}{A_v}.$$

Onde define-se

$$\iint_{A_v} \int_{z_{min}}^{z_{max}} q(r, \theta, z) dz r dr d\theta \equiv Q_v,$$

o valor total de produção calorífica na vareta v , que pode ser obtida pela soma dos *tallies* de produção calorífica de cada umas das células que compõem a região do cerne combustível da vareta (pois assume-se, conforme já mencionado, que o calor é gerado apenas nesta região). Deste modo, ao se integrar $q_v(z) = \beta_v \bar{q}(z)$ entre z_{min} e z_{max} nos dois lados da equação, tem-se:

$$\int_{z_{min}}^{z_{max}} q_v(z) dz = \beta_v \left[\frac{1}{N} \sum_p \int_{z_{min}}^{z_{max}} q_p(z) dz \right] \Rightarrow \frac{Q_v}{A_v} = \beta_v \left[\frac{1}{N} \sum_p \frac{Q_p}{A_p} \right],$$

e, assim, ao considerar, como já assumido anteriormente, que a área de seção transversal A_v é a mesma para todas as varetas combustíveis, β_v pode ser encontrado para uma vareta combustível v qualquer ao se fazer

$$\beta_v = \frac{Q_v}{\frac{1}{N} \sum_p Q_p},$$

onde o somatório percorre todas as varetas combustíveis, N é o número de varetas combustíveis e, conforme mencionado, Q_v é o valor total de calor produzido na vareta v , que pode ser obtido diretamente por *tallies* sobre as células correspondentes ao cerne combustível da vareta. Com a

obtenção de tal relação, pode-se então calcular o fator β_R da vareta representativa (que produziu a maior quantidade de calor no núcleo) da seguinte forma:

$$\beta_R = \frac{Q_R}{\frac{1}{N} \sum_p Q_p} = \frac{\max_p \{Q_p\}}{\frac{1}{N} \sum_p Q_p},$$

e, portanto, finalmente, pode-se encontrar a distribuição axial de densidade volumétrica de produção calorífica da vareta representativa, $q_R(z)$:

$$q_R(z) = \beta_R \bar{q}(z) = \beta_R \frac{Q(z)}{A} = \left[\frac{\max_p \{Q_p\}}{\frac{1}{N} \sum_p Q_p} \right] \frac{Q(z)}{A}.$$

Após a obtenção de $q_R(z)$, que está originalmente nas unidades de elétron-Volt por partícula-fonte.cm³, para se encontrar a densidade volumétrica de potência, $\dot{q}_R(z)$ em unidades de [W/cm³], multiplica-se $q_R(z)$ que está em [eV/(pf.cm³)] pelo fator f já mencionado que converte [eV/pf] para [J/s], resultando em [J/(s.cm³)] \equiv [W/cm³], isto é, $\dot{q}_R(z) = f q_R(z) = f \beta_R \frac{Q(z)}{A}$.

Também pode-se calcular facilmente o valor do fator α_v para cada vareta v : visto que tal fator é a razão entre a produção calorífica da vareta v e a produção calorífica da vareta representativa, tem-se, portanto:

$$\alpha_v \equiv \frac{Q_v}{Q_R} = \frac{Q_v}{\max_p \{Q_p\}}$$

onde p percorre o conjunto de todas as varetas combustíveis.

Com a obtenção do perfil axial de densidade volumétrica de potência da vareta representativa, $\dot{q}_R(z) = f q_R(z) = f \beta_R \frac{Q(z)}{A}$, pode-se calcular a distribuição de temperatura $T^R(\vec{r})$ do conjunto vareta representativa-LBE circundante através do modelo de difusão-convecção. Tal modelo é aplicado no domínio espacial da Figura 3.16, correspondente ao conjunto supramencionado. Deve ser observado que, dado que se utilizará apenas a distribuição axial de densidade de potência $\dot{q}_R(z)$ como termo fonte do modelo, o domínio correspondente à vareta/LBE

é axissimétrico e portanto bidimensional nas coordenadas cilíndricas (r, z) . Deste modo, a equação de difusão-convecção em tal domínio é dada por:

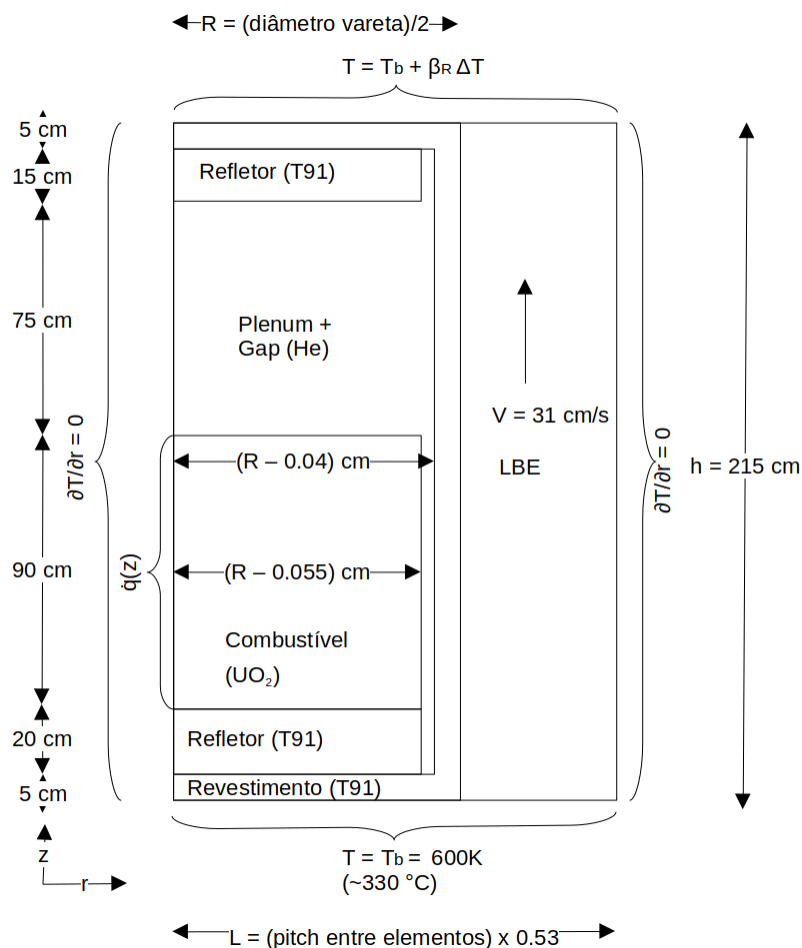
$$\begin{aligned} -\nabla \cdot (\lambda \nabla T^R) &= \dot{q}_R && \text{na vareta combustível,} \\ -(\lambda + \lambda_T) \nabla^2 T^R + \rho C_p \vec{v} \cdot \nabla T^R &= 0 && \text{na região do fluido refrigerante,} \end{aligned}$$

com $\vec{v} = [v_r \ v_z]$, $v_r = 0$, $v_z = 0,31 \text{ m/s} = 31 \text{ cm/s}$ e $\dot{q}^R(\vec{r}) \equiv \dot{q}^R(z)$ diferente de zero apenas na região do cerne combustível. Aqui, $\lambda + \lambda_T \equiv \lambda_{eff}$, é o valor de condutividade térmica efetiva, sendo a soma da condutividade térmica molecular λ e λ_T , a condutividade térmica advinda do fluxo turbulento no LBE. Este último termo é definido por $\lambda_T = (C_p \mu_T) / (Pr)_T$, sendo μ_T a viscosidade dinâmica turbulenta e $(Pr)_T$ o número de Prandtl turbulento. Estes dois parâmetros podem ser obtidos por relações semi-empíricas: $(Pr)_T = 4.12$ [11], sendo válido especificamente para o LBE com números de Péclet ($Pe = Pr \cdot Re$) menores que 1000, o que corresponde ao regime do SPARK-NC, enquanto $\mu_T = 0.0123 Re^{0.854} \mu$ [12], válido para fluidos em geral, incluindo metais líquidos, com $2100 \leq Re < 3.6 \times 10^6$, $0.0107 \leq Pr \leq 5.65$, estando também dentro da faixa de valores do regime. Deve ser observado que o modelo não contempla uma camada limite de velocidades na interface vareta/LBE; porém, visto que o LBE possui um número de Prandtl baixo ($Pr \approx 0.02$) e o regime possui um número de Reynolds relativamente reduzido ($Re \approx 15000$), tais efeitos possuem baixo impacto na distribuição de temperatura, pois a camada limite de velocidade terá pouca influência sobre a camada limite térmica [9].

Conforme pode ser observado na Figura 3.16, o domínio tem $z = h = 215 \text{ cm}$ na direção z , correspondente à altura da vareta, e largura L na direção r , que é igual a $\sqrt{\sqrt{3}/2\pi}$ (≈ 0.53) vezes o valor do *pitch* entre duas varetas no elemento, de modo a manter a área transversal de fluxo de LBE do domínio cilíndrico equivalente à área transversal de fluxo do prisma hexagonal da região original. Visto que o *pitch* das varetas varia a depender da região do núcleo (ver Tabela 3.1), o valor é ajustado de acordo no momento do ajuste termohidráulico com a região onde a vareta representativa é encontrada. O mesmo acontece para o raio R da vareta, que pode diferir entre regiões e portanto é ajustada no momento do cálculo da mesma forma. Observa-se também na figura que o raio interno da vareta e o raio do combustível/refletor é baseado no valor de R , subtraídos, respectivamente, da espessura do revestimento e da espessura do revestimento + espessura do *gap*, que possuem o mesmo valor para as varetas de qualquer região do núcleo. O número de elementos de volume na simulação é de 215 elementos na direção z e 200 elementos na direção r , resultando em uma malha retangular de 43000 elementos no total.

Nota-se também que tal abordagem assume que a região vareta/LBE correspondente ao domínio é considerado como um canal segregado em que não ocorre troca térmica com a vizinhança, em oposição a cálculos de subcanal em que se contemplam trocas laterais de *momentum* e de calor. Dado que a transferência de calor no LBE é fortemente condutiva e que o fluxo não é laminar, em um primeiro momento a abordagem parece não ser adequada; porém, ao se invocar a já mencionada hipótese de que a distribuição radial de potência no núcleo varia suavemente na vizinhança de uma dada vareta, que o formato da distribuição axial das varetas são similares, e que a velocidade \vec{v} do LBE é aproximada como uniforme sobre o núcleo, é esperado que a distribuição de temperatura seja similar nas regiões circunvizinhas, e portanto os efeitos da difusão térmica radial podem ser desconsiderados.

Figura 3.16 – Domínio da vareta representativa/LBE circundante e condições de contorno empregadas na simulação numérica para o cálculo da distribuição de temperatura no domínio.



Conforme pode ser visto na Figura 3.16, as condições de contorno de temperatura são as seguintes:

- $\partial T(r, z)/\partial r = 0$ em $r = 0$, devido a condição de simetria rotacional no centro do conjunto vareta/LBE;
- $\partial T(r, z)/\partial r = 0$ em $r = L$. Tal condição também é de simetria, presumindo que a região vareta representativa/LBE é cercado de regiões vareta/LBE circunvizinhas que possuem a mesma distribuição de temperatura e, como tal condição será mapeada para todas as outras varetas combustíveis, assume-se também que o mesmo ocorre para todas elas. No entanto, isso pode ser considerado uma boa aproximação se ao se invocar a supramencionada hipótese de que a distribuição de temperatura seja similar nessa vizinhança;
- $T_R(r, 0) = T_b = 600K \sim 330^\circ C$, pois é a temperatura de entrada do LBE (ver Tabela 3.4), e considera-se que toda a região inferior esteja em equilíbrio térmico;
- $T_R(r, h) = T_b + \beta_R \Delta T$. β_R é, conforme já discutido, a razão entre a produção calorífica na vareta representativa e a produção calorífica média tomada entre as varetas combustíveis e ΔT é a elevação média de temperatura no núcleo, que é obtida pelo balanço de energia que relaciona tal elevação com a potência térmica gerada no núcleo, o calor específico do fluido, e o fluxo de massa do fluido em todo o núcleo. Pela Tabela 3.4 $\dot{m} = 1746 \text{ kg/s}$, $P = 30 \text{ MWt}$ e a capacidade térmica aproximada do LBE em torno da temperatura de entrada é de $C_p \simeq 143 \text{ J/(kg. K)}$ [6], então tem-se :

$$\Delta T = \frac{P}{\dot{m}C_p} = \frac{3 \times 10^7 [J/s]}{143 [J/(kg.K)] \cdot 1746 [kg/s]} = 120.15 \text{ K}.$$

A imposição desta última condição faz com a média, tomada sobre todas as varetas combustíveis, das elevações de temperatura ΔT_v da região de cada vareta combustível v , dada por $\Delta T_v \equiv T_v(r, h) - T_v(r, 0) = T_v(r, h) - T_b = \alpha_v(T_R(r, h) - T_b) = \alpha_v \beta_R \Delta T$, seja igual a ΔT , o que é uma boa aproximação ao se considerar que o fluxo de massa \dot{m}_v que percorre cada vareta combustível v possui o mesmo valor para todas as varetas:

$$\overline{\Delta T} = \frac{1}{N} \sum_v \Delta T_v = \frac{1}{N} \sum_v \alpha_v \beta_R \Delta T = \frac{1}{N} \sum_v \left[\frac{Q_v}{\max_p \{Q_p\}} \right] \left[\frac{\max_p \{Q_p\}}{\frac{1}{N} \sum_p Q_p} \right] \Delta T = \Delta T.$$

Conforme mencionado, o modelo permite que os coeficientes variem não só com a posição, mas também com a temperatura: assim, o modelo é mais realístico ao passo que também torna-se não-linear, exigindo portanto um método iterativo para a sua solução. As equações que relacionam as propriedades termohidráulicas de todos os materiais com a temperatura são dadas abaixo; deve

ser notado que C_p e ρ são especificados apenas para o LBE pois é a única região do domínio onde $\vec{v} \neq 0$. Nessas relações, T é em unidades de $[K]$, λ é dado em $[W/(m.K)]$, C_p em $[J/(kg.K)]$ e ρ em $[kg/m^3]$, que são apropriadamente convertidos para unidades baseadas em centímetros antes da inserção no modelo:

- $\lambda_{UO_2}(T) = 7.6 - 0.0076(T - 273.15) + 0.0000026(T - 273.15)^2$. É a aproximação quadrática, válida na faixa $300\text{ K} < T < 1800\text{ K}$, para a fórmula clássica [13] de condutividade térmica do UO_2 com porosidade de 95% (portanto próxima à porosidade utilizada neste trabalho, ver Tabela 3.5):

$$\lambda_{UO_2} = \frac{100}{7.5408 + 17.692\tau + 3.6142\tau^2} + \frac{6400}{\tau^{5/2}} \exp\left(\frac{-16.35}{\tau}\right), \quad \tau = T/1000$$

- $\lambda_{T91}(T) = 24.66 + 0.015(T - 273.15) - 0.000011(T - 273.15)^2$. É uma aproximação quadrática contínua para a relação discreta (não aceita como *input* pelo FiPy), encontrada em [14]:

$$\lambda_{T91} = \begin{cases} 26 \Rightarrow (T - 273.15 \leq 50); & 27 \Rightarrow (50 < T - 273.15 \leq 150); \\ 28 \Rightarrow (150 < T - 273.15 \leq 300); & 29 \Rightarrow (300 < T - 273.15 \leq 450); \\ 30 \Rightarrow (T - 273.15 > 450). \end{cases}$$

- $\lambda_{LBE}(T) = 3.61 + (1.517 \times 10^{-2})T - (1.741 \times 10^{-6})T^2$. É a expressão recomendada em [6];
- $\rho_{LBE}(T) = 11096 - 1.3236T$. É a expressão recomendada em [6] (ver Tabela 3.5);
- $C_p^{(LBE)} = 159 - (2.302 \times 10^{-2})T$. É a expressão recomendada em [6];
- $\lambda_{He} = (2.63 \times 10^{-3})T^{0.71}$. Relação clássica entre temperatura e condutividade térmica do Hélio, utilizado como modelo simplificado em códigos de cálculo de desempenho de combustível como o BISON [15].

Para o *plenum*, considera-se apenas que o espaço é preenchido por um material estático com a condutividade dada por λ_{He} ; não são levados em consideração fenômenos convectivos dentro do *plenum* típicos ao caso em questão. Porém, essa é uma abordagem utilizada em modelos simplificados de códigos como o TRANSURANUS [16]. Para o *gap*, os efeitos convectivos também são desconsiderados e, adicionalmente, os efeitos de radiação também são desprezados: Para temperaturas típicas de operação, a condutividade aproximada devida à radiação tem pequena importância relativa, exceto possivelmente em varetas com uma grande espessura de *gap* e emissão

de larga quantidade de produtos de fissão gasosos, que degradam a condutividade térmica do material original de preenchimento do *gap* [17].

A condutividade térmica original do *gap* é modificada nas interfaces *gap*-combustível e *gap*-revestimento: em tais interfaces, utiliza-se no lugar da condutividade do Hélio, a média harmônica das condutividades entre os dois lados da interface [18]:

$$\lambda_{gap-UO_2} = \frac{2\lambda_{He}\lambda_{UO_2}}{(\lambda_{He} + \lambda_{UO_2})}, \quad \lambda_{gap-T91} = \frac{2\lambda_{He}\lambda_{T91}}{(\lambda_{He} + \lambda_{T91})}.$$

Portanto, as interfaces recebem como valor de condutividade modificada λ_{gap-UO_2} e $\lambda_{gap-T91}$, e a parte central do *gap* recebe o valor puro da condutividade do Hélio, λ_{He} .

Por fim, a distribuição de densidade volumétrica de potência $\dot{q}_R(z)$ é inserida na região do material combustível dentro do domínio do modelo: subdivide-se a região em n “faixas” de altura $\Delta_z = h_{UO_2}/n$, onde h_{UO_2} é a altura do cerne combustível na vareta, e amostra-se n pontos de $\dot{q}_R(z)$ dentro da região do combustível, atribuindo o valor referente a cada ponto à sua sub-região correspondente. Aqui mais uma vantagem das expansões funcionais fica evidente: dado que $\dot{q}_R(z)$ é uma função contínua, o número de pontos amostrados é arbitrário, e portanto pode-se aproximar a distribuição axial de densidade de potência na vareta com precisão arbitrária no modelo. De posse da definição do domínio e das condições de contorno, das propriedades dos materiais componentes e da distribuição de potência que é dada como entrada do modelo, o mesmo pode ser solucionado para a obtenção da distribuição espacial de temperatura $T^R(\vec{r})$ na vareta representativa, que então é mapeada em cada célula c dessa vareta utilizando a média dos valores de $T^R(\vec{r})$ na região ocupada em cada célula, obtendo-se então T_c^R pela relação

$$T_c^R = \frac{1}{V_c^R} \iiint_{V_c^R} d^3r T^R(\vec{r})$$

Sendo V_c^R o volume da célula c da região da vareta representativa, sendo tal integral discretizada de acordo com a malha do domínio. Assim, da posse de α_v para todas as varetas combustíveis v e T_c^R , pode-se calcular a temperatura de cada célula correspondente à essas varetas, isto é, T_c^v .

A relação dada acima que utiliza o fator α_v para o cálculo das temperaturas das células de uma dada região é válida somente para regiões de varetas combustíveis, que geram potência calorífica e para o qual se pode obter \dot{q}_v . No entanto, deve-se atribuir temperaturas para as outras regiões do núcleo, como os elementos contendo varetas de blindagem/refletoras e dos elementos de

controle/SCRAM, além dos elementos completamente preenchidos por LBE mostrados na Seção 3.1.2. Para os elementos contendo regiões de varetas de blindagem/refletoras, além dos elementos puramente LBE, nota-se que os mesmos formam um “anel” externo em torno dos elementos centrais, que são compostos quase que exclusivamente por elementos combustíveis (ver Figura 3.17 (a)). Neste caso, por não haver fontes significativas de geração de potência nesses elementos fez-se uma abordagem puramente condutiva: Considerou-se o problema clássico da condução de calor em um anel cilíndrico sem fontes de calor, com a parede interna do anel estando à uma temperatura T_i , enquanto a temperatura da parede externa é dada por T_o (ver Figura 3.17 (b)).

Observa-se na que, se somente a condução for considerada, pode-se tratar, numa aproximação simplificada de fluido estagnante, a distribuição de temperatura na região desses elementos de maneira similar à este problema, onde T_i é uma distribuição de temperatura dada pela temperaturas dos revestimentos dos elementos combustíveis mais externos e T_o é a temperatura da parede do vaso do reator, que, devido à estar afastado das fontes de geração de calor por fissão do centro do núcleo, é tomado como estando em equilíbrio térmico com a temperatura de entrada do LBE, isto é, $T_o = T_b \approx 600\text{K}$. O problema é então modelado da seguinte maneira:

$$\frac{1}{r} \frac{d}{dr} \left[\lambda(r, T) r \frac{dT(r)}{dr} \right] = 0, \quad T(r_i) = T_i, \quad T(r_o) = T_o$$

Uma vantagem desse problema é que, ao se assumir que a condutividade térmica λ nessa região é constante ele possui uma solução analítica que independe das propriedades térmicas do material no interior da casca cilíndrica, dependendo apenas das temperaturas T_o e T_i e dos raios interno e externo [19]:

$$T(r) = T_i + \frac{T_o - T_i}{\ln(r_o/r_i)} \ln \left(\frac{r}{r_i} \right)$$

Conforme mencionado, a temperatura T_i é extraída das temperaturas dos revestimentos dos elementos combustíveis do anel mais externo dentre esses elementos. Porém, devido ao fato que pode haver uma diferença nas temperaturas dos elementos ao longo desse anel, ele foi dividido em setores circulares (ver Figura 3.18), onde T_i foi considerada como a temperatura média dos revestimentos dos elementos de cada setor e todos os elementos de blindagem, refletores e de LBE que estão dentro desse setor têm a temperatura dada conforme a solução do problema de casca cilíndrica, com r_i o raio aproximado do anel mais externo dentre os elementos combustíveis e r_o a distância média entre o elemento cuja temperatura deseja se calcular e os elementos combustíveis

do anel mais externo que estão naquele setor circular. Deve ser observado que tal solução é apenas uma aproximação, pois λ não é estritamente homogênea na região considerada e o fato de se considerar a distribuição de T através de setores circulares faz com que T dependa também de θ e portanto $\partial T / \partial \theta \neq 0$, fazendo com que a equação do modelo não seja somente radial. Não obstante, o modelo pode ser considerado como razoável se as variações de λ e $\partial T / \partial \theta$ forem pequenas na região devido à sua linearidade.

Observa-se também que apenas uma temperatura é dada por elemento, portanto tal temperatura é atribuída a todas as células dos elemento (incluindo as células de LBE). Por este motivo, dentre os elementos de blindagem, refletores e de LBE não há subdivisão detalhada de células, e cada componente desses elementos é composto por apenas uma célula.

Figura 3.17 - (a) – Região dos elementos refletores/blindagem/LBE em um anel concêntrico à região central dos elementos combustíveis no núcleo. (b) Problema clássico de condução de calor em uma região anular.

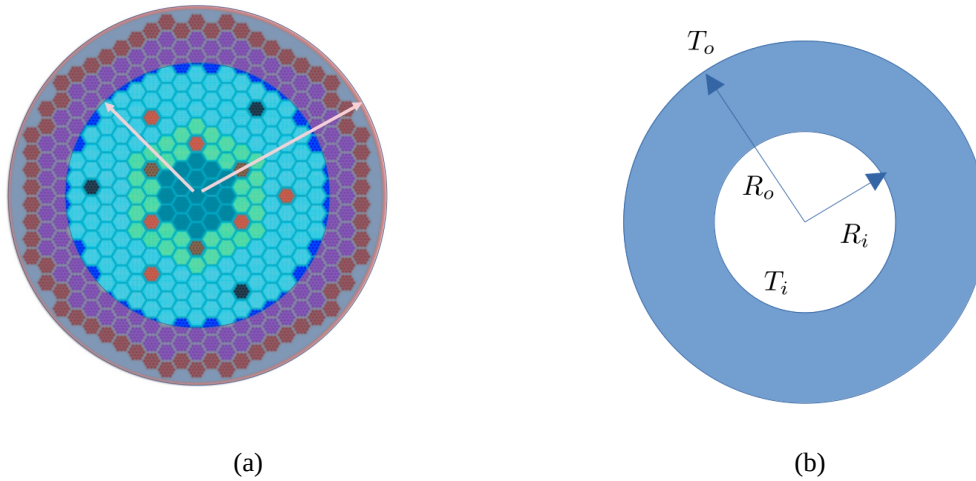
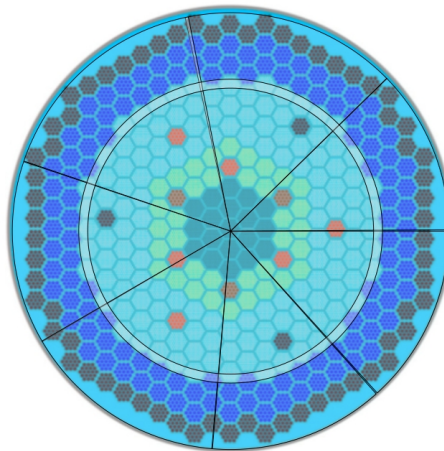
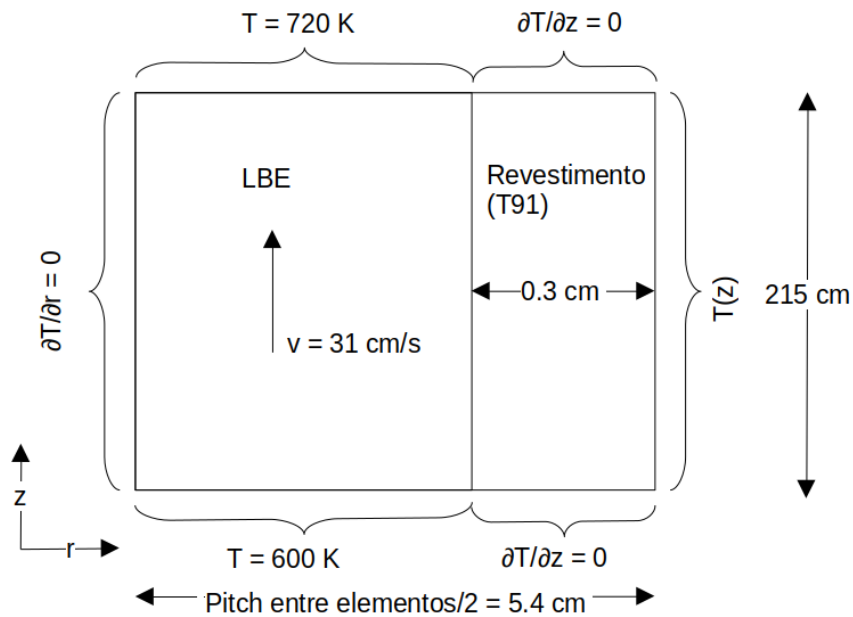


Figura 3.18 – Divisão do núcleo em setores circulares para o cálculo de T_i .



Os elementos de controle/SCRAM permanecem totalmente levantados do núcleo nos cálculos de criticalidade e depleção (As grandezas em [1] são calculadas com o núcleo no estado “ARO” - *all rods out*, com todas as barras extraídas). Deste modo, a geometria do núcleo durante as simulações tem apenas o espaço de inserção desses elementos, que é preenchido por LBE. Tais espaços ficam completamente envolvidos por elementos combustíveis, mas não geram potência calorífica como esses elementos. O autor optou por fazer uma simulação numérica da distribuição de temperaturas nesses espaços, atribuindo as temperaturas às células correspondentes. Visto que as propriedades dos materiais e as equações do modelo permanecem inalteradas, será mostrado apenas o domínio onde o cálculo é feito e as condições de contorno: também utiliza-se a simetria axial com domínio bidimensional em (r, z) , de altura $h = 215$ cm e largura igual à metade do *pitch* entre elementos, ou seja, $L = 5.4$ cm, o que faz com que, no domínio do modelo, o elemento hexagonal seja transformado em um cilindro equivalente com espessura de valor igual ao revestimento da estrutura fixa hexagonal do elemento (0.3 cm). No eixo central ($r = 0$) impõe-se a condição de axissimetria ($\partial T / \partial r = 0$), enquanto em $r = L$ impõe-se o perfil axial de temperatura $T(z)$ tomado como a média dos perfis axiais dos revestimentos dos elementos combustíveis que envolvem o elemento em questão. Na região do LBE, tem-se $T = 600$ K em $z = 0$ cm e $T = 720$ K em $z = 215$ cm, além de $v_z = 31$ cm/s e $v_r = 0$ cm/s, enquanto na região correspondente ao revestimento impõe-se a condição de isolamento ($\partial T / \partial z = 0$) em $z = 0$ cm e $z = 215$ cm, com $v_z, v_r = 0$ cm/s. O domínio é mostrado na Figura 3.19.

Figura 3.19 – Domínio para o cálculo de temperatura nos espaços de inserção dos elementos de controle/SCRAM.



3.2.2 A modelagem do processo de ajuste termohidráulico

Esta seção mostrará alguns aspectos do processo de modelagem em código do ajuste termohidráulico. Dado que é um processo longo que foi descrito com relativa riqueza de detalhes na subseção 3.2.1, a discussão irá ater-se aos seus dois componentes principais: A obtenção de $Q(z)$, a densidade linear axial de produção calorífica na região ativa do núcleo e Q_v , a produção calorífica total em cada vareta combustível v , dos quais se obtêm β , α_v e $\dot{q}_R(z)$ conforme mostrado na Subseção 3.2.1, e a solução numérica do modelo de difusão-convecção para a obtenção da distribuição de temperatura da vareta representativa, $T^R(\vec{r})$, através de $\dot{q}_R(z)$. Conforme já mencionado na subseção 3.1.2, as funções relativas ao ciclo termohidráulico estão no módulo **SPARKThermal**.

Conforme mencionado na Subseção 3.2.1, o OpenMC oferece funções *built-in* para a obtenção de *tallies* por expansão funcional. A distribuição $Q(z)$ pode ser obtida ao se criar um filtro que conjuga dois sub-filtros: Um filtro que denota a expansão por polinômios de Legendre, e outro que denota a região cilíndrica onde se fará a expansão, que, para o problema em questão, tem um raio R que abrange todos os elementos combustíveis e altura igual à altura do cerne combustível das varetas. Para o ajuste termohidráulico neste trabalho, utiliza-se a ordem da expansão de Legendre $N_z = 8$ para a obtenção de $Q(z)$:

```
ordem_leg = 8 #expansão em polinômios de Legendre de ordem 8
pb = openmc.Zplane(z_min) #planos em z= z_min e z=z_max
pt = openmc.Zplane(z_max)
cyl = openmc.Zcylinder(r=R) #cilindro de raio R
regiao = -cyl & +pb & -pt #regiao cilindrica
cilindro = openmc.CylindricalMesh.from_domain(domain=regiao,dimension=[1,1,1])

#criacao do filtro de Legendre
filtro_Legendre = openmc.SpatialLegendreFilter(ordem_leg, 'z', z_min,z_max)
tally = openmc.Tally() #criacao do tally
tally.filters = [openmc.MeshFilter(cilindro), filtro_Legendre] #atribuicao dos filtros
tally.scores = ['heating-local'] #atribuição do contabilizador (score): calor gerado
```

Os tallies relativos aos coeficientes da expansão podem ser recuperados durante a simulação e atribuídos ao polinômio $P(z)$ através de uma função já existente no OpenMC, gerando $Q(z)$:

```
coeficientes_leg = openmc.lib.tallies[tally_ID].mean #recupera os coeficientes a_n

#produz Q_z atraves da expansao com os coeficientes
```

```
Q_z = openmc.legendre_from_expcoef(coeficientes_leg, domain=(z_min,z_max))
```

Os valores de Q_v , isto é, a produção calorífica total em cada vareta v , podem ser obtidos através de *tallies* diretos sobre cada uma das células que compõem a região do cerne combustível uma vareta v , que são então somados para que se tenha a produção calorífica total na vareta.

De posse de $Q(z)$ e de Q_v para cada vareta v , pode obter-se, conforme mostrado na subseção 3.2.1, β , α_v e a distribuição axial da densidade volumétrica de potência da vareta representativa, $\dot{q}_R(z)$, que é então alimentado no módulo de cálculo numérico termohidráulico para a obtenção da distribuição espacial de temperatura no conjunto vareta/LBE circundante representativo, $T^R(\vec{r})$. A partir de $T^R(\vec{r})$, pode-se obter, através de um mapeamento que utiliza a média de $T^R(\vec{r})$ nas regiões das células, a temperatura nas células desse conjunto, T_c^R , que então é utilizada para atualizar a temperatura de todas as células associadas à vareta v por meio de $T_c^v = T_b + \alpha_v(T_c^R - T_b)$ em cada ciclo de interrupção/atualização, com a posterior atualização das temperaturas das células restantes do núcleo conforme mostrado na Subseção 3.2.1. A implementação em *Python* do módulo de cálculo numérico termohidráulico será discutido na subseção seguinte.

3.2.2.1 – O cálculo da distribuição de temperatura pela biblioteca FiPy

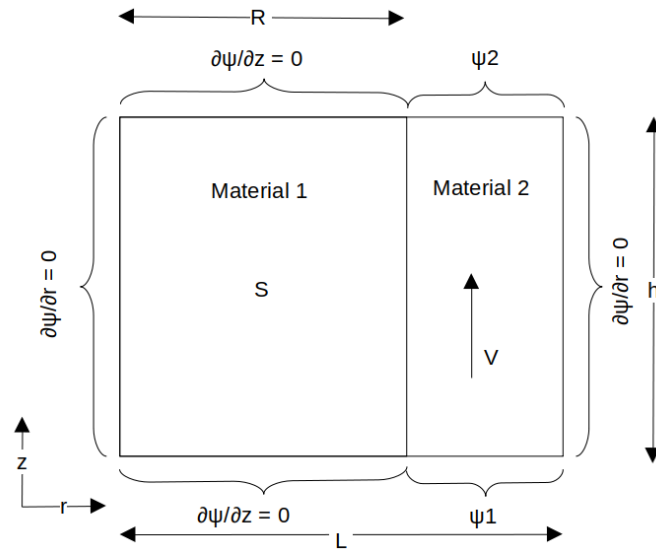
Para o cálculo da distribuição de temperatura $T^R(\vec{r})$ a partir de $\dot{q}_R(z)$ e das condições de contorno no domínio axissimétrico correspondente à vareta representativa/LBE circundante especificadas na subseção 3.2.1, o autor utilizou a biblioteca do *Python* **FiPy** [20], desenvolvida por pesquisadores do NIST (*National Institute of Standards and Technology*), nos Estados Unidos. A biblioteca oferece funções que permitem a solução por volumes finitos de uma gama extensa de equações diferenciais parciais elípticas, parabólicas e hiperbólicas com termos transientes, difusivos, e convectivos de diversas ordens com fontes externas arbitrárias, também permitindo que os coeficientes dos termos sejam variantes no tempo e no espaço e dependentes, de maneira linear ou não-linear, da variável a ser solucionada. Além da versatilidade oferecida, a biblioteca oferece a vantagem de ser gratuita e poder ser invocada diretamente no ambiente *Python* onde o OpenMC está sendo executado, permitindo uma integração facilitada entre os cálculos neutrônicos e termohidráulicos.

De modo a exibir um panorama das funcionalidades do FiPy, será mostrado o código que fornece a solução, em uma geometria simples, de um problema de difusão-convecção em estado estacionário com fonte externa e coeficientes variantes no espaço e dependentes de uma variável

escalar qualquer Ψ a ser solucionada, pois é conceitualmente menos complexa que o problema de difusão-convecção do ajuste termohidráulico, ao mesmo tempo que apresenta similaridades com ele: a extensão do código mostrado abaixo para o modelo do ajuste termohidráulico, tomando-se Ψ como a variável T , é relativamente simples e portanto não há perda de generalidade.

O domínio-exemplo, em coordenadas cilíndricas axissimétricas (r, z) é mostrado na Figura 3.20. Ele tem raio L e altura h , e é dividido em dois materiais distintos, 1 e 2, ambos com altura h e raios R e $L - R$, respectivamente. Nas fronteiras esquerda e direita do domínio tem-se $\frac{\partial \Psi}{\partial r} = 0$, enquanto nos planos superior e inferior do Material 1 tem-se $\frac{\partial \Psi}{\partial z} = 0$. No plano inferior do Material 2 tem-se $\Psi = \Psi_1$, enquanto no plano inferior o valor é Ψ_2 . O Material 2 é um fluido e move-se à velocidade v na direção z , enquanto o Material 1 é um sólido possuindo uma fonte S (que para o caso térmico seria a densidade volumétrica de potência \dot{q}). Os coeficientes difusivos dos materiais 1 e 2 são $\lambda_1(\Psi) = a\Psi$ e $\lambda_2(\Psi) = b\Psi^2$, respectivamente, e o coeficiente de convecção do Material 2 é dado por $\sigma(\Psi) = c\Psi^3$.

Figura 3.20 – Domínio-exemplo para a explanação das funções do FiPy.



A equação no domínio é dada então por um termo difusivo, um convectivo e uma fonte:

$$-\nabla \cdot (\lambda(\vec{r}, \Psi) \nabla \Psi(\vec{r})) + \nabla \cdot (\sigma(\vec{r}, \Psi) \vec{v} \Psi(\vec{r})) = S(\vec{r}),$$

$$\text{com} \quad \lambda(\vec{r}, \Psi) = \begin{cases} a\Psi, & r \leq R \\ b\Psi^2, & r > R \end{cases} \quad \vec{v} = \begin{cases} [0, 0], & r \leq R \\ [0, v], & r > R \end{cases} \quad \sigma(\vec{r}, \Psi) = \begin{cases} 0, & r \leq R \\ c\Psi^3, & r > R \end{cases}$$

$$S(\vec{r}) = \begin{cases} S, & r \leq R \\ 0, & r > R \end{cases} \quad \left(\frac{\partial \Psi}{\partial r} \right)_{r=0, L} = 0, \quad \left(\frac{\partial \Psi}{\partial z} \right)_{z=0, h}^{r \leq R} = 0, \quad \Psi|_{z=0}^{r > R} = \Psi_1, \quad \Psi|_{z=h}^{r > R} = \Psi_2.$$

Inicia-se a solução do problema no FiPy pela definição do domínio cilíndrico axissimétrico através do objeto `CylindricalGrid2D` e pelo refinamento da malha (espaçamento em r e z dos elementos de volume, que são, respectivamente dr e dz), e pela definição da variável a ser solucionada Ψ , que pode ser inicializada com um valor arbitrário:

```
from fipy import CellVariable, CylindricalGrid2D, TransientTerm, DiffusionTerm,
FaceVariable, PowerLawConvectionTerm

#cria domínio cilíndrico axissimétrico – nx e ny é o numero de elementos de volume na
horizontal e na vertical
dominio = CylindricalGrid2D(dx=dr, dy=dz, nx=L/dr, ny=h/dz)
# cria variável a ser solucionada, dando a ela o valor inicial V_inic
Psi = CellVariable(name = "Psi", mesh = dominio, value = V_inic)
```

Posteriormente, cria-se as regiões distintas que compreendem os dois materiais, utilizando-se como condição para a seleção da região a posição no eixo horizontal dos elementos de volume da malha, que é dado pelo atributo `x` do objeto correspondente ao domínio. Feito isso, associa-se o valor da condutividade térmica e coeficiente de convecção apropriados a cada região, somando-os para obter os valores totais em todo o domínio: a soma neste caso corresponde a uma união disjunta das regiões com as propriedades materiais associadas a cada uma delas. Deve ser notado que as propriedades dos materiais são dependentes da variável a ser solucionada, Ψ .

```
material1 = dominio.x <= R #definicoes das regioes dos materiais 1 e 2
material2 = dominio.x > R

coef_diff_1 = a*Psi*material1 #atribuicoes das propriedades dos materiais às regiões
coef_diff_2 = b*(Psi**2)*material2
coef_conv_1 = 0*material1
coef_conv_2 = c*(Psi**3)*material2

# obtencao das propriedades difusivas e convectivas em todo o domínio
coef_diff = coef_diff_1 + coef_diff_2
coef_conv = coef_conv_1 + coef_conv_2
```

Em seguida, as condições de contorno devem ser impostas nas fronteiras do domínio. Para as fronteiras da esquerda e da direita, que tem a mesma condição $\frac{\partial \Psi}{\partial r} = 0$ em toda a sua extensão, pode-se utilizar os atributos `facesLeft` e `facesRight` do domínio, impondo o gradiente nulo através do atributo `faceGrad.constrain` da variável Ψ . Para as fronteiras superior e inferior as condições são distintas a depender do material, e portanto regiões são criadas nessas fronteiras, para que as condições de contorno possam ser impostas separadamente. Para isto, utiliza-se a função

`faceCenters`, que retorna as posições em r e z de todos os elementos de volumes da malha do domínio:

```
#imposição das cond. contorno nos limites esquerdo e direito
Psi.faceGrad.constrain(0,where=dominio.facesLeft)
Psi.faceGrad.constrain(0,where=dominio.facesRight)

rf , zf = dominio.faceCenters #retorna as coordenadas de todas as faces dos elementos
do dominio
#imposicao das cond. contorno nos limites sup. e inf. no material 1
Psi.faceGrad.constrain(0, where= (dominio.facesTop)&(rf <= R))
Psi.faceGrad.constrain(0, where= (dominio.facesBottom)&(rf <= R))

#imposicao das cond. contorno nos limites sup. e inf. no material 2
Psi.constrain(T2, where=(dominio.facesTop)&(rf > R))
Psi.constrain(T1, where=(dominio.facesBottom)&(rf > R))
```

A fonte externa S é modelada na região correspondente ao Material 1 ($r < R$):

```
Fonte = S*material1
```

Após a especificação do domínio, dos coeficientes, das condições de contorno e da fonte externa, a equação pode ser definida. A equação deve estar na forma canônica, isto é, com todos os termos em um só lado da equação e o outro lado igual a zero. A função `DiffusionTerm(coeff=lambda)` corresponde à $\nabla \cdot (\lambda(\vec{r}, \Psi) \nabla \Psi)$, a função `PowerLawConvectionTerm(coeff=sigma*[vr,vz])` corresponde à $\nabla \cdot (\sigma \vec{v} \Psi)$ com $\vec{v} = [v_r, v_z]$. Portanto, para a equação do modelo, tem-se:

```
eq = DiffusionTerm(coeff=cond_termica) - PowerLawConvectionTerm(coeff=coef_conv*[0,V])
+ Fonte
```

Visto que os coeficientes dependem da própria variável a ser solucionada, exige-se um método iterativo para a solução da equação. O FiPy permite que isso seja feito pela função `sweep` (varredura), que resolve a equação no número de vezes que o usuário especificar através de um `loop for`, atualizando a variável Ψ a cada iteração:

```
NUM_ITERACOES = 10

#resolve a equação para a variável Psi iterativamente, retornando o resíduo (diferença
entre a solucao da iteração anterior e a da atual) a cada iteração
```

```
for i in range(NUM_ITERACOES):
    residuo = eq.sweep(var=Psi)
```

Finalmente, a variável já solucionada pode ser convertida para uma matriz de formato (n_r, n_z) , onde $n_r = L/dr$, $n_z = h/dz$, obtendo-se efetivamente $\Psi(\vec{r})$ para o domínio em questão:

```
Dist_Psi = Psi.value.reshape(nz,nr).tolist() # matriz Psi(nz,nr) com os valores
```

O módulo termohidráulico resolve a equação similar à mostrada neste exemplo com $\Psi = T$ mas com o domínio da vareta representativa mostrado na Figura 3.20 e os coeficientes e condições de contorno especificados na subseção 3.2.1, além de uma fonte de densidade de potência não-uniforme $S = \dot{q}_R(z)$ no combustível (o que pode ser conseguindo ao dividir a região do combustível em sub-regiões, por meio dos mesmos artifícios utilizados no exemplo). Assim, consegue-se uma matriz numérica correspondente à $T^R(\vec{r})$, que pode ser mapeada através de médias nas regiões associadas às células do conjunto da vareta representativa, obtendo-se então T_c^R para cada uma dessas células.

3.2.3 – Descrição resumida do estágio de ajuste termohidráulico

Esta subseção oferece uma descrição resumida do estágio de ajuste termohidráulico:

1. Cria-se e inicializa-se o objeto núcleo, e dá-se valores iniciais às temperaturas/densidades dos materiais nas células;
2. Executa o cálculo de criticalidade por um número N de bateladas de partículas, de modo a gerar estatística suficiente para a geração dos *tallies*;
3. Interrompe-se o cálculo de criticalidade, e dá-se início ao primeiro ciclo de ajuste termohidráulico. Faz-se a leitura dos *tallies*. A partir dessas leituras, obtém-se os valores do fator de conversão f (que converte de $[eV/pf]$ para $[J/s]$, onde pf é partícula-fonte), os valores de produção calorífica total em cada vareta v e a densidade linear axial de produção calorífica total $Q(z)$ na região das varetas combustíveis, sendo essa última advinda de uma expansão funcional contínua;

4. Calcula-se o perfil axial de densidade de potência para uma vareta representativa $\dot{q}_R(z)$, que é a vareta que produziu a maior potência neste ciclo, através da relação $\dot{q}_R(z) = f q_R(z) = f \beta_R \bar{q}(z) = f \beta_R \frac{Q(z)}{A}$. $\beta_R = \frac{Q_v}{\frac{1}{N} \sum_p Q_p}$, onde Q_v é a produção calorífica total da vareta v mencionado na Etapa 3, A é a soma das áreas de seção reta de todas as varetas combustíveis e f é o fator de conversão também mencionado na Etapa 3 que converte de densidade volumétrica de produção calorífica em $[eV/(pf.cm^3)]$ para $[J/(s.cm^3)] \equiv [W/cm^3]$;
5. O valor de $\dot{q}_R(z)$ é inserido em um módulo termohidráulico, onde é calculada no FiPy a distribuição de temperatura $T^R(\vec{r})$ na região da vareta representativa e do LBE circundante à mesma, assumindo um modelo de difusão-convecção, domínio axissimétrico e condições de contorno/coeficientes apropriados, conforme mostrado na subseção 3.2.1;
6. A partir de $T^R(\vec{r})$, obtém-se a temperatura T_c^R de cada célula da vareta representativa, fazendo-se um mapeamento onde a célula recebe o valor de temperatura da média de $T^R(\vec{r})$ na região correspondente à célula;
7. Obtém-se, para cada vareta combustível v , o fator α_v , que relaciona a produção calorífica em v com a produção calorífica na vareta representativa, isto é: $\alpha_v = \frac{Q_v}{Q_R} = \frac{Q_v}{\max_p \{Q_p\}}$;
8. A temperatura numa célula c associada à região de uma dada vareta v (incluindo o LBE que circunda essa vareta) é obtida através da temperatura da célula c correspondente na região da vareta representativa através da relação $T_c^v = T_b + \alpha_v(T_c^R - T_b)$, onde $T_b = 600K$ é a temperatura mínima esperada no núcleo;
9. De posse da temperatura atualizada das células do LBE nas regiões das varetas, atualiza-se também a sua densidade por meio da relação $\rho_{LBE}(T) = 11096 - 1.3236T$ (em kg/m^3 , que deve ser convertido apropriadamente para g/cm^3) [6];
10. Atualiza as temperaturas das células do revestimento dos elementos combustíveis: para um dado elemento, este valor é a média das temperaturas das células de LBE que circundam todas as varetas dentro desse elemento;

11. Calcula-se a temperatura das células dos elementos refletores e de blindagem, além do LBE que circunda o conjunto total de elementos do núcleo, através da relação mostrada em 3.2.1; as densidades das células preenchidas por LBE nesses elementos também são atualizadas conforme a relação mostrada no passo 9;
12. Calcula-se a temperatura das células nos espaços das barras levantadas de controle/SCRAM que estão preenchidos por LBE. Tal temperatura também é advinda de um cálculo termohidráulico, onde há o mapeamento de temperatura das células através das médias das temperaturas nas regiões correspondentes;
13. Após a atualização das temperaturas de todas as células e das densidades das células de LBE, o ciclo de ajuste termina, e executa-se mais um número N' de bateladas, até um novo ciclo de interrupção e ajuste ou até terminar a simulação de criticalidade caso tenha atingido convergência ou o número máximo especificado de bateladas. Na opção de se iniciar um novo ciclo de interrupção/ajuste, faz-se novamente os passos 3 a 12.
14. Ao término da simulação, coleta-se o valor de k_{eff} com seu desvio padrão, também armazenando a distribuição de temperatura/densidade no núcleo encontrada no último ciclo em variáveis denominadas `CELL_TEMPERATURES` e `MATERIAL_DENSITIES`, de modo a carregá-la nas células do novo objeto núcleo que será utilizado para o estágio de depleção subsequente (ver o fluxograma no início do Capítulo 3 e a Seção 3.3).

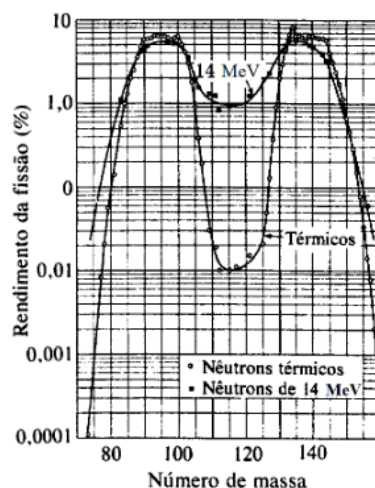
3.3 – O estágio de depleção

Esta seção descreve o estágio de depleção realizado no modelo do SPARK-NC no OpenMC. É apresentada uma breve introdução sobre o processo de análise de depleção em reatores, seguida da descrição do código e os dados de entrada utilizados para a sua execução.

3.3.1 – A descrição do processo de análise de depleção

A análise de depleção simula a evolução temporal do estado de composição isotópica do núcleo, que é induzida pela queima de seu material combustível. A realização dos processos de fissão necessários para a geração de potência durante a operação do reator resulta em geral na criação de dois núcleos-filhos, cuja distribuição em termos de número de massa varia amplamente (ver Figura 3.21). Adicionalmente, esses núcleos-filhos gerados podem se envolver em reações de captura ou serem instáveis, decaindo de acordo com séries predefinidas, com os tempos de meia-vida associados à tais decaimentos diferindo entre si por várias ordens de grandeza. Deste modo, a população de núclídeos distintos daqueles encontrados nos materiais com que o reator foi construído tende a aumentar conforme o tempo de operação.

Figura 3.21 - Probabilidade de ocorrência dos produtos de fissão (Rendimento de fissão) de acordo com o número de massa para a fissão do ^{235}U por nêutrons térmicos e de 14 MeV. Retirado de [21].



O objetivo da análise de depleção é determinar a quantidade de cada um desses produtos de fissão e seus subprodutos de transmutação no núcleo em pontos temporais específicos do ciclo de vida do reator, através de equações que relacionam a evolução da quantidade de um certo núclídeo com parâmetros relacionados aos processos de produção dos núcleos-filhos e seus respectivos

decaimentos (rendimentos de fissão, razões de ramificação, meias-vidas associadas, etc.). Adicionalmente, a análise determina, em cada um desses pontos temporais, a criticalidade do núcleo através do fator de multiplicação efetivo (k_{eff}), visto que a mesma não se mantém constante: além da queima diminuir a concentração de material físsil, o que diminui a probabilidade de fissão e portanto o valor de k_{eff} , os produtos de fissão gerados ao longo do tempo possuem propriedades que introduzem uma dinâmica complexa na neutrônica do núcleo; por exemplo, reatores térmicos podem produzir uma grande quantidade de Xenônio-135, como produto direto de fissão ou por decaimento β^- a partir do núcleo-filho Telúrio-135. O ^{135}Xe possui uma seção de choque de captura de cerca de 2.5 milhões de *barns*, sendo o “veneno nuclear” mais poderoso conhecido, e tem meia-vida relativamente curta (cerca de 9.5 horas). A eliminação de ^{135}Xe , tornando-se por captura de nêutrons o isótopo estável ^{136}Xe , depende do fluxo instantâneo de nêutrons, enquanto a produção de ^{135}Xe depende do histórico do fluxo integrado no tempo, o que pode causar oscilações de concentração de Xenônio de curto período que afetam a criticalidade do núcleo (por outro lado, reatores rápidos operam em um alto fluxo de nêutrons, fazendo com que o ^{135}Xe seja rapidamente transmutado em ^{136}Xe e não acumule de forma significativa no núcleo).

A variação temporal da densidade N_i do nuclídeo i num volume V onde assume-se que essa densidade não varia espacialmente é dada pelas *equações de Bateman* :

$$\begin{aligned} \frac{dN_i(t)}{dt} = \sum_j & \underbrace{\left[\underbrace{\frac{1}{V} \sum_c \int_V d^3r \int_0^\infty dE f_{j \rightarrow i}^c(E) \sigma_j^c(\vec{r}, E, t) \phi(\vec{r}, E, t)}_{\text{outras transmutações (c = fissão, captura, etc.)}} + \underbrace{\lambda_{j \rightarrow i}}_{\text{decaimento}} \right]}_{\text{Produção do nuclídeo i pelo nuclídeo j} \equiv \Lambda_{j \rightarrow i} N_j(t)} N_j(t) \\ & - \underbrace{\left[\underbrace{\frac{1}{V} \sum_c \int_V d^3r \int_0^\infty dE \sigma_i^c(\vec{r}, E, t) \phi(\vec{r}, E, t)}_{\text{outras transmutações (c = fissão, captura, etc.)}} + \underbrace{\sum_{j \neq i} \lambda_{i \rightarrow j}}_{\text{decaimento}} \right]}_{\text{Perda do nuclídeo i} \equiv \Lambda_{i \rightarrow *} N_i(t)} N_i(t) \end{aligned}$$

Nesta equação, $\phi(\vec{r}, E, t)$ é o fluxo neutrônico, $f_{j \rightarrow i}^c(E)$ é a fração de reações, para uma dada energia E de nêutron incidente e de um dado canal c de transmutação exceto decaimento (fissão, captura, etc.) do nuclídeo j que produzem o nuclídeo i , σ_i^c é a seção de choque microscópica para o canal c de transmutação do nuclídeo i , no ponto \vec{r} , na energia E e no tempo t , e $\lambda_{j \rightarrow i}$ é a constante de decaimento para todos os modos possíveis em que o nuclídeo j decai para o

nuclídeo i . Nota-se que as equações acima assumem que o balanço de todos os nuclídeos que entram e saem do volume de controle V é nulo. Dado que geralmente V é tomado como a região correspondente ao cerne combustível das varetas e a maioria dos nuclídeos produzidos tendem a permanecer na estrutura cristalina do combustível sólido, tal hipótese é adequada e as equações de depleção como mostradas acima geralmente oferecem uma boa aproximação. Alguns produtos de fissão gasosos como o supramencionado ^{135}Xe tendem a se difundir da matriz do combustível para o *gap/plenum*, mas, devido à modelagem de tais eventos envolver processos adicionais complexos de transporte de partículas, esses eventos geralmente não são considerados em códigos de transporte de nêutrons como o OpenMC ou o MCNP. Entretanto, deve ser mencionado que para o cálculo de depleção em combustíveis líquidos (como aqueles em reatores baseados em sais fundidos), pode-se inserir nas equações acima um termo de transferência que modela a retirada/inserção de nuclídeos de um volume fixo [22].

As equações de Bateman podem ser escritas na seguinte forma matricial, para os n nuclídeos que têm a possibilidade de serem gerados no interior do núcleo durante um processo de depleção:

$$\begin{bmatrix} dN_1/dt \\ dN_2/dt \\ \vdots \\ dN_i/dt \\ \vdots \\ dN_n/dt \end{bmatrix} = \begin{bmatrix} -\Lambda_{1 \rightarrow *} & \Lambda_{2 \rightarrow 1} & \cdots & \cdots & \cdots & \Lambda_{n \rightarrow 1} \\ \Lambda_{1 \rightarrow 2} & -\Lambda_{2 \rightarrow *} & \cdots & \cdots & \cdots & \Lambda_{n \rightarrow 2} \\ \vdots & & \ddots & & & \\ \Lambda_{1 \rightarrow i} & \Lambda_{2 \rightarrow i} & \cdots & -\Lambda_{i \rightarrow *} & \cdots & \Lambda_{n \rightarrow i} \\ \vdots & & & & \ddots & \\ \Lambda_{1 \rightarrow n} & \Lambda_{2 \rightarrow n} & \cdots & \Lambda_{i \rightarrow n} & \cdots & -\Lambda_{n \rightarrow *} \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_i \\ \vdots \\ N_n \end{bmatrix} \Rightarrow \begin{aligned} \frac{d\mathbf{N}(t)}{dt} &= A(\phi, \mathbf{N}, t)\mathbf{N}(t) \\ \mathbf{N}(0) &= \mathbf{N}_0 \end{aligned}$$

Na notação compacta acima, \mathbf{N}_0 é o vetor contendo a quantidade inicial de cada nuclídeo no início da análise de depleção. Deve ser notado que os parâmetros Λ dependem do fluxo ϕ , que por sua vez é obtido de maneira direta através da solução das equações de transporte, que depende da densidade dos nuclídeos. Desta maneira, $\Lambda = \Lambda(\phi, N_1, N_2 \dots N_n)$ e, portanto, a matriz A depende do vetor \mathbf{N} , tornando a equação não-linear e exigindo um método iterativo para sua solução. O método mais simples (método preditor) encontra aproximações sucessivas para a matriz A (chamada de matriz de queima ou de *burnup*) utilizando a aproximação de \mathbf{N} de uma iteração anterior, substituindo-a na solução formal para a equação, que é $\mathbf{N} = \mathbf{N}_0 e^{At}$, $e^{At} \equiv \sum_{n=0}^{\infty} \frac{A^n t^n}{n!}$, de modo a encontrar o valor de \mathbf{N} para a iteração atual. O OpenMC oferece, além do método preditor, vários outros métodos para a solução numérica da equação diferencial matricial acima, com diversos graus de sofisticação que aceleram a convergência e a precisão dos resultados, ao passo que incorrem em custos associados de memória e tempo de processamento.

A solução formal da equação diferencial matricial é relativamente simples e exige apenas esforço de cálculo numérico, dado que a matriz de *burnup* A esteja definida nos intervalos de tempo em que o cálculo é feito. Entretanto, visto que $A(\phi, \mathbf{N}, t)$ depende do fluxo ϕ no instante t , a solução do sistema de equações matriciais necessita uma simulação de transporte de partículas no núcleo que forneça esse fluxo para o cálculo de \mathbf{N} em um dado instante t . Tal valor de \mathbf{N} , por sua vez é usado para atualizar a concentração de núclídeos no reator, atualizando-o para uma nova iteração da simulação de transporte que irá fornecer o valor do fluxo no instante $t + \Delta t$, e assim sucessivamente até atingir convergência. Deste modo, observa-se que as soluções de transporte e de depleção devem estar acopladas.

As soluções das equações acopladas mencionadas acima baseiam-se em um conjunto de dados denominado *arquivo de cadeia de depleção (depletion chain file)*, que deve incluir as seguintes informações:

- Para cada núclídeo, quais canais de transmutação são possíveis, os núclídeos que são os produtos finais de tais transmutações e o valor de Q (isto é, a energia liberada ou absorvida) de cada uma das reações. Tais dados são necessários para os valores de $f_{i \rightarrow j}^c$ e $\lambda_{i \rightarrow j}$ para qualquer par de núclídeos (i, j) .
- Se o núclídeo não é estável, quais os modos de decaimento são possíveis, suas razões de ramificação (*branching ratios*) e seus produtos de decaimento, com suas respectivas meias-vidas. Tais dados são necessários para os valores de $\lambda_{i \rightarrow j}$ para qualquer par de núclídeos (i, j) .
- Se um núclídeo é fissionável, o rendimento de fissão para valores de energias de nêutrons incidentes. Tais dados são necessários para os valores de $f_{i \rightarrow j}^c(E)$, onde c é o canal de fissão, para qualquer par de núclídeos (i, j) .

Na solução das equações de depleção acopladas com a simulação de transporte, deve-se calcular as taxas de reação – ou seja, as integrais dependentes do fluxo neutrônico nas equações de Bateman - que serão inseridas na equação matricial de modo a se calcular a variação temporal da densidade dos núclídeos. Por serem dependentes do fluxo, essas taxas são calculadas como um *tally* regular de uma grandeza no método Monte Carlo, e como tal são dadas em reações por partícula-fonte. De modo a converter para reações por segundo, que é a unidade das grandezas a serem alimentadas na equação matricial para que se tenha o resultado de dN/dt em núclídeos/(cm³.s),

deve-se utilizar um fator de conversão que leva em consideração a potência do reator, como aquela realizada para o *tally* de calor gerado nas varetas na Subseção 3.2.1. Assim, a potência de operação do reator (que pode ser constante ou variar ao longo das etapas do cálculo) é um dado de entrada importante na análise de depleção. Também deve ser mencionado que o acoplamento transporte/depleção calcula, através do valor da evolução temporal do fluxo neutrônico do núcleo durante a depleção, o valor de k_{eff} no início e no fim da faixa temporal especificada para a análise.

Por fim, discute-se sobre o tamanho do passo de depleção (1 ano), que pode, em um primeiro momento, parecer demasiado longo, fazendo com que se perca informação importante na dinâmica neutrônica do núcleo; no entanto, uma regra geral utilizada [23] para a escolha do tamanho do passo temporal na depleção é utilizar passos de queima de 2MW-dia por kg de metal pesado, sendo a massa tomada como o inventário total de metal pesado do núcleo no início da operação em $t = 0$ (se o reator for projetado com veneno queimável tal passo de queima deve ser de apenas 1MWd/kg). Tal valor pode ser calculado como:

$$\frac{P \times \Delta}{M_{hm}} = 2 \left[\frac{MWd}{kg} \right]$$

sendo Δ o passo máximo em dias, P a potência do reator e M_{hm} a massa inicial de metal pesado. Ao substituir os valores de potência e de massa inicial de metal pesado conforme a Tabela 3.4, tem-se:

$$\Delta = 2 \left[\frac{MWd}{kg} \right] \times \frac{M_{hm}}{P} = 2 \left[\frac{MWd}{kg} \right] \times \frac{5745kg}{30MW} = 383d$$

onde pode-se observar que o valor máximo permitido de passo temporal é maior que 365 dias, portanto 1 ano é considerado um passo adequado para a depleção. Adicionalmente, o fato do SPARK-NC ser um reator rápido faz com que a simulação não necessite de passos menores de depleção para que sejam capturados efeitos dinâmicos que ocorrem em períodos relativamente curtos, como as oscilações de Xenônio típicas de reatores térmicos.

3.3.2 – A modelagem da análise de depleção do SPARK-NC no OpenMC

A modelagem de uma análise de depleção no OpenMC é relativamente simples, pois grande parte das funções necessárias para a análise, que foram descritas na Subseção 3.3.1, estão

encapsuladas pelo código, cabendo ao usuário definir alguns parâmetros e dados de entrada. Esta seção tem por objetivo mostrar como o estágio de depleção para o núcleo do SPARK-NC foi implementado no OpenMC. O estágio é feito através de uma função denominada **depletionStep** que está definida no módulo **SPARKDepletion**, que foi mencionado na Subseção 3.1.2. A função **depletionStep** recebe como dados de entrada, dentre outros, os seguintes mais importantes:

- O arquivo com a temperatura de cada célula do modelo de núcleo do SPARK-NC que foi criado para o estágio de ajuste termohidráulico imediatamente anterior ao estágio de depleção. Deve ser lembrado que todo estágio de depleção entre $t = N$ anos e $t = N + 1$ anos é antecedido por um estágio de ajuste termohidráulico que define a temperatura de estado estacionário do núcleo no ano N , conforme foi mencionado no início do Capítulo 3. Tais temperaturas serão mapeadas para o modelo de núcleo do SPARK-NC (com menor refinamento em termos de número de células) que será empregado para o cálculo de depleção;
- O arquivo com todas as densidades de cada material, que preenche cada célula, do modelo de núcleo do SPARK-NC que foi criado para o estágio de ajuste termohidráulico imediatamente anterior ao estágio de depleção. Nota-se que, se o estágio de ajuste termohidráulico em questão não é aquele relativo a $t = 0$, isto é, se já houve um estágio de depleção anterior, os materiais que preenchem cada célula associada ao combustível incluirão nuclídeos distintos daqueles pertencentes à composição inicial do núcleo em $t = 0$, e portanto a densidade dos materiais combustíveis são modificadas devido a mudanças na composição dos materiais. Do mesmo modo, tais densidades serão apropriadamente mapeadas para o modelo de núcleo do SPARK-NC com menor granularidade que será empregado para o cálculo de depleção.
- O caminho para o arquivo de cadeia de depleção mencionado anteriormente, fornecendo os dados necessários para o cálculo das taxas de reação e constantes de decaimento das equações de Bateman.

Inicialmente, a função **depletionStep** cria um objeto **CoreMaterials** contendo todos os materiais necessários para o preenchimento das células que compõem a geometria do núcleo em que a análise de depleção será feita. Como se deseja um núcleo com menor refinamento em número de

células, um conjunto reduzido de materiais deve ser criado, e portanto passa-se uma variável booleana como argumento (`MULTIPLE_MATERIALS`) sinalizando tal desejo na instanciação do objeto:

```
import SPARKUtilities as su
coremats = su.CoreMaterials(MULTIPLE_MATERIALS =False)
```

Após a criação do objeto com o repositório de materiais, cria-se o objeto referente ao núcleo. Passa-se como argumento, dentre outros dados menos importantes, o repositório de materiais, a indicação de que se deseja um núcleo com menor refinamento de células através da variável `MULTIPLE_MATERIALS` e um dicionário (`SLICES_DICT`) que indica em quantas células cada um dos componentes do núcleo deve ser subdividida: naturalmente, tal número de subdivisões deve ser menor que o número de subdivisões dos componentes para o modelo de núcleo com maior granularidade. O dicionário `SLICES_DICT` é definido no módulo **SPARKData**.

```
import SPARKData as sd

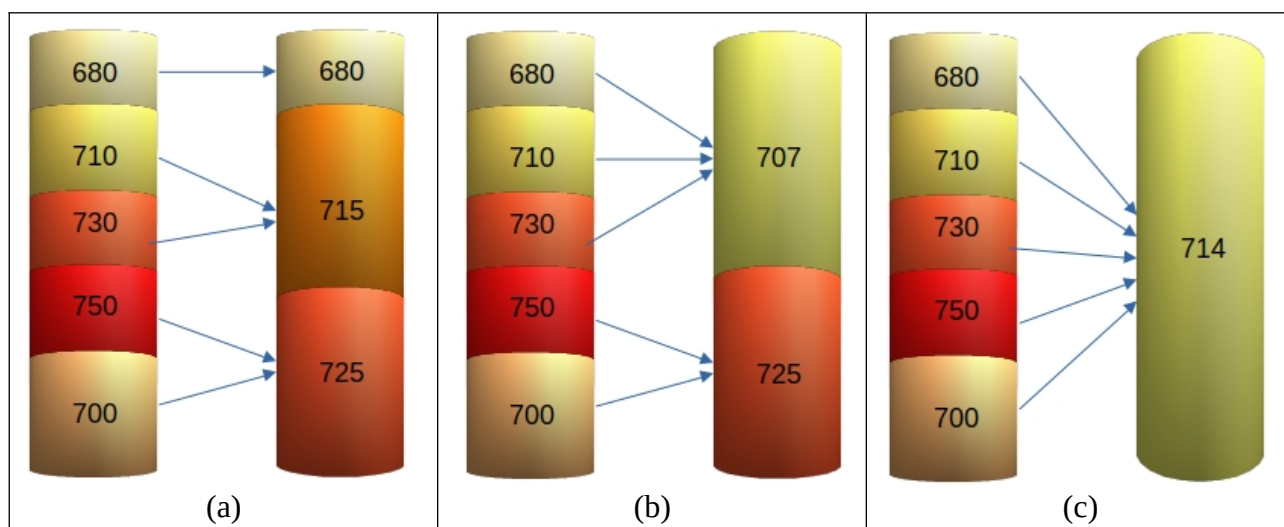
core=su.ReactorCore(coremats =coremats,SLICES_DICT =sd.SLICES_DICT,MULTIPLE_MATERIALS
=False, ...)
# os ... indicam outros argumentos menos importantes
```

Findada a criação do núcleo, deve-se carregá-lo com as temperaturas e densidades advindas da etapa de ajuste termohidráulico anterior. Dado que o núcleo nessa etapa possui uma maior granularidade de células, deve-se fazer um mapeamento de tais grandezas para um conjunto menor de células pertencente ao modelo do núcleo para depleção.

Conforme mostrado na Subseção 3.2.3, após o estágio de ajuste termohidráulico são disponibilizados duas variáveis de dados físicos: `_CELL_TEMPERATURES` e `_MATERIAL_DENSITIES`. Tais variáveis são gravadas em arquivos que ficam na memória principal do ambiente do *python*, e são resgatadas durante o estágio de depleção de modo a serem mapeadas para o modelo de núcleo deste estágio. Para a temperatura, isto é feito através da função **updateCoreTemp** do módulo **SPARKUtilities**. A função recebe como argumento o modelo do núcleo para o estágio de depleção e o arquivo `_CELL_TEMPERATURES`, e mapeia entre a distribuição de temperaturas de uma dada instância de um dado componente no modelo de núcleo com maior refinamento para a distribuição de temperatura da mesma instância desse componente no modelo de núcleo com menor refinamento através de “médias localizadas”, onde o valor de temperatura atribuída a uma dada célula é uma média aritmética das temperaturas das células do modelo mais refinado que possuem uma região em comum com a célula, conforme pode ser visto na Figura 3.22. Para o estágio de depleção, o autor

optou por um núcleo simplificado com apenas uma célula por componente devido o custo computacional associado ao cálculo de depleção; assim, tal célula recebe a média aritmética de todas as células associadas ao componente no núcleo correspondente ao estágio de ajuste termohidráulico, conforme mostrado na Figura 3.22 (c).

Figura 3.22 - Mapeamento de temperatura das células entre o mesmo componente do SPARK-NC, porém em modelos com granularidades distintas. O componente do modelo com maior granularidade possui 5 células, e mostra-se o mapeamento para um componente no modelo de menor granularidade com (a) 3 células, (b) 2 células e (c) 1 célula. Nota-se que atribui-se uma média aritmética, ponderada pelo volume relativo de cada subdivisão, da temperaturas das células que estão numa região correspondente à célula que terá o valor de temperatura atribuído.



A função **updateCoreTemp** retorna após a sua execução um dicionário contendo as temperaturas calculadas para cada uma das células do modelo de depleção, que é então atribuído ao objeto correspondente ao núcleo de depleção com o auxílio de funções subsidiárias do OpenMC, definindo completamente sua distribuição de temperatura conforme herdada do estágio de ajuste termohidráulico imediatamente anterior:

```
tempdict = su.updateCoreTemp(core, _CELL_TEMPERATURES)
for j in tempdict:
    core.universe.get_all_cells()[j].temperature = tempdict[j]
```

A densidade é mapeada de uma maneira mais simples que a temperatura: As densidades de materiais não-LBE permanecem inalteradas (pois são consideradas imutáveis ao longo de toda a simulação, conforme mencionado na Subseção 3.1.2.2), enquanto as densidades dos materiais LBE são atribuídas através de um mapeamento direto, visto que existe apenas um material para cada região preenchida por LBE no modelo com maior granularidade (também mencionado em 3.1.2.2).

Deste modo, o mapeamento é feito simplesmente atribuindo a densidade do material de uma cada célula de uma dada região, encontrada para o núcleo de ajuste termohidráulico, a uma célula da mesma região no núcleo de depleção. Isso é feito através da variável `_MATERIAL_DENSITIES`, que contém a densidade de cada célula associada com uma região preenchida por LBE.

O código de **depletionStep** cria então um objeto da classe **Model**, que, no OpenMC, é uma espécie de “*container*” que encapsula não só a geometria e os materiais de um dado problema, mas também as configurações necessárias para a execução de uma simulação de transporte de partículas. Alguns exemplos não-exaustivos de parâmetros de configuração são:

- O tipo de simulação: cálculo de criticalidade, simulação com fonte fixa de partículas, cálculo estocástico de volume, modo gráfico, etc.;
- O número de bateladas ativas e inativas de cálculo e o número de histórias de partículas por batelada;
- Como será o cálculo da seção de choque para uma dada temperatura da célula: interpolação das seções de choque relativas às temperaturas próximas à célula ou se a seção de choque relativa à temperatura mais próxima à da célula é considerada, além das faixas de tolerância relacionadas aos dois métodos;
- Se fótons devem ser considerados na simulação (por *default*, somente nêutrons são considerados);
- definição da fonte de partículas inicial (em caso de cálculo de criticalidade) ou fixa para o problema em questão.

As configurações de uma simulação Monte Carlo no OpenMC podem ser criadas através de um objeto da classe **Settings**. A simulação de transporte para a depleção contém 1200 bateladas de cálculo, sendo 70 bateladas inativas e 35000 histórias de partículas por batelada, portanto o código possui as seguintes linhas:

```
settings = openmc.Settings()
settings.particles = 35000
settings.batches = 1200
settings.inactive = 70
```

Para se definir a fonte de partículas, criou-se uma caixa paralelepipedica que envolve todo o núcleo, com uma distribuição uniforme de pontos que denota a localização das fontes. Porém, de modo a acelerar a convergência do cálculo, e para se evitar a tomada de pontos na região da caixa

que estão fora do núcleo, deseja-se ter pontos de fonte apenas no espaço preenchido por materiais combustíveis. Para isto, a definição de fontes oferece um parâmetro chamado `only_fissionable`:

```
epsilon=0.1
TOTAL_RADIUS = sd.CORE_RADIUS+sd.CORE_WALL_THICKNESS+epsilon
criticality_source = openmc.stats.Box([-TOTAL_RADIUS, -TOTAL_RADIUS, 0 ],
[TOTAL_RADIUS, TOTAL_RADIUS, sd.TOP_CORE+epsilon], only_fissionable=True)
settings.source = openmc.IndependentSource(space=criticality_source)
```

As seções de choque para uma célula a uma dada temperatura deverão ser dadas por interpolações de tabelas de seções de choque calculadas para temperaturas fixas próximas à temperatura da célula, portanto tem-se no código:

```
settings.temperature = {
    'default': sd.STANDARD_TEMP,
    'method': 'interpolation',}
```

Nas linhas acima, *'default'* é a temperatura padrão que deve ser atribuída a uma célula caso ela não tenha temperatura definida no código.

A partir da geometria e dos materiais presentes do núcleo, pode-se criar objetos das classes do OpenMC **Geometry** e **Materials** (não confundir com a classe **Material** mostrada na Subseção 3.1.2, que compreende a definição de um único material), que são objetos que devem ser passados como argumento para a criação do objeto da classe **Model**:

```
geom = [core.cells[0]]
for i in range(len(core.cells[1])):
    geom.append(core.cells[1][i])
geometry = openmc.Geometry(geom)

materials = openmc.Materials(core.materials)
```

Com isso, o objeto pode finalmente ser criado, ao se passar os objetos de geometria, materiais e configurações de simulação:

```
model = openmc.Model(geometry=geometry, settings=settings, materials=materials)
```

É necessário que se informe ao OpenMC quais materiais são considerados depletáveis para a presente análise, isto é, em quais materiais se deseja acompanhar a cadeia de transmutação dos

nuclídeos a partir da fissão desses materiais. Tal informação é dada pelo parâmetro booleano `depletable` da classe **Material**. Como deseja-se acompanhar a depleção de todo o material combustível do núcleo, este parâmetro deve estar sinalizado como verdadeiro nos três materiais que preenchem os três tipos de combustível (pertencentes aos anéis interno, médio e externo do núcleo). Desta forma, durante a instanciação do objeto da classe **ReactorCore**, ou seja, o objeto que representa o núcleo, tal atributo é colocado como verdadeiro para esses materiais:

```
self.fuelin.depletable=True
self.fuelmid.depletable=True
self.fuelout.depletable=True
```

A análise também necessita da massa total de material depletável, e como apenas as densidades dos materiais são atribuídas nas definições dos materiais, o volume total das células contendo os materiais depletáveis devem ser fornecidos através do parâmetro `volume` da classe **Material**. Isto também é feito durante a criação do repositório dos materiais pelo objeto **CoreMaterials**: quando da criação dos materiais combustíveis pela função **setMaterials** (ver a Subseção 3.1.2.2), o volume total daquele material é calculado através dos parâmetros geométricos das varetas e da quantidade de elementos e varetas por elemento que contêm o material, sendo então associado à variável `volume` daquele material:

```
if FISSILE:
    mat.volume = num*npins*math.pi*pow(diam/2,2)*height
```

Conforme mencionado na Subseção 3.1.2.3, quando os dois níveis de reticulado hexagonal são criados (reticulado de varetas no interior do elemento e reticulado de elementos no interior do núcleo), as células definidas para um dado componente do elemento (p.ex. revestimento do elemento, componentes da vareta do elemento, LBE circundante às varetas) são replicadas e transladadas a partir da origem do seu sistema de coordenadas onde a célula foi criada, sendo tal translação feita de acordo com a posição que o componente ocupa no sistema de coordenadas do reticulado. Deste modo, a definição de um ou mais níveis de reticulado cria várias “instâncias” das células, sendo o número de instâncias igual ao número de vezes em que esse componente se repete, quando se considera todos os níveis.

Caso uma análise de depleção for iniciada sem que nenhuma informação adicional em relação à que foi feita até então no código for fornecida, o OpenMC compreenderá que todo o material depletável está totalmente localizado na região onde a célula original foi definida, isto é, no sistema

de coordenadas com origem no centro do reator $\vec{r} = (0, 0, 0)$. Assim, os cálculos de transporte acoplados considerarão os efeitos neutrônicos referentes apenas àquela localização, incorrendo em uma perda considerável de informação sobre a física do problema, e também atualizará a composição isotópica do material apenas no material daquela célula original, o que não reflete o processo físico real da queima do combustível no núcleo. Entretanto, de modo a fornecer ao OpenMC a informação de que o material está distribuído em todo o reticulado, a classe **Model** oferece a função `differentiate_depletable_mats`, que cria diferentes materiais na memória interna do OpenMC, distribuindo o volume total do material depletável em cada instância da célula que é replicada em todos os pontos pertinentes do núcleo, e realizando a depleção em cada material separadamente conforme o fluxo neutrônico localizado na região onde a instância da célula se encontra. Assim, o código também deve possuir as seguintes linhas:

```
model.differentiate_depletable_mats(diff_volume_method = 'divide equally')
```

Assim, o OpenMC identifica automaticamente as células com materiais depletáveis e a localização das instâncias de cada uma dessas células, atribuindo um novo material para cada uma das células com volume dividido igualmente entre elas (por isso o argumento `divide_equally` na função), também informando ao cálculo de transporte que, para o novo material referente à uma dada célula, a depleção deve considerar o fluxo neutrônico na região da instância daquela célula.

Caso o estágio de depleção seja aquele referente à faixa $[t = 0, t = 1 \text{ ano}]$ isto é, seja a primeira etapa de depleção, a composição de materiais é a da especificação original do reator, conforme em [1]. Porém, se o estágio corresponder a uma faixa temporal posterior, o objeto do núcleo deve, além das temperaturas e densidades advindas da etapa de ajuste termohidráulico anterior, também as composições isotópicas dos materiais depletáveis obtidas no fim do estágio de depleção imediatamente anterior ao estágio de depleção onde o código se encontra (ver o fluxograma na Figura 3.2). Ao final de cada estágio de depleção, o código gera um arquivo no ambiente *Python* onde guarda-se uma variável denominada `_DEPL_MAT`, que contém um dicionário de materiais depletáveis de cada instância de cada célula pertinente com a composição isotópica atualizada. No estágio de depleção seguinte, tal variável é recuperada e carregada no modelo corrente do núcleo:

```
cont_mat = 0
for i in range(len(model.materials)):
    if (model.materials[i].depletable == True):
        model.materials[i] = _DEPL_MAT[model.materials[i].name][cont_mat]
```



```
cont_mat += 1
if (cont_mat == len(_DEPL_MAT[model.materials[i].name])):
    cont_mat = 0
```

Deve ser mencionado que, após o fim do estágio corrente de depleção, `_DEPL_MAT` é atualizada com a composição isotópica referente ao fim daquele estágio de queima, sendo gravada novamente em arquivo.

O objeto da classe **Model** tem então suas informações de geometria, materiais, e configurações de simulação exportadas para arquivos `.xml`, de modo a permitir a execução do código Monte Carlo pelo módulo principal do OpenMC:

```
model.export_to_xml()
```

Então, cria-se um objeto da classe **CoupledOperator**, que é o operador de transporte acoplado à solução das equações de Bateman, conforme mostrado em 3.3.1:

```
operator = openmc.deplete.CoupledOperator(model, DEPLETION_CHAIN_PATH)
```

A instanciação de tal objeto toma como argumento o objeto da classe **Model** e o caminho para o arquivo `.xml` (`DEPLETION_CHAIN_PATH`) que possui os dados do supramencionado arquivo de cadeia de depleção, necessário para a solução das equações acopladas. Existem vários arquivos de depleção disponíveis no sítio do OpenMC [24], sendo que o módulo de depleção do OpenMC também oferece funções que permitem a geração desses arquivos a partir de bibliotecas de dados nucleares. O autor escolheu, dentre as diversas opções no sítio do OpenMC, o arquivo simplificado de cadeias de transmutação aplicado ao espectro rápido, abrangendo isótopos recomendados em [25] e razões de ramificação de reações de captura de nêutrons típicas de um reator rápido refrigerado à sódio líquido. Tal arquivo é menos completo que outros disponíveis; por exemplo, há arquivos de cadeia de espectro rápido baseados em toda a gama de isótopos das bibliotecas ENDF-B/VII.1 e ENDF-B/VIII.0. Porém, a escolha do autor foi feita de modo a tentar obter resultados relativamente satisfatórios com uma demanda computacional menos intensa, pois o tempo de cálculo e uso de memória crescem significativamente com a complexidade das cadeias de transmutação e o número de isótopos envolvidos.

A classe **CoupledOperator** contém o atributo `heavy_metal`, mencionado na Subseção 3.1.2.2 que calcula, através das composições isotópicas dos materiais depletáveis e dos volumes declarados para as células que são preenchidos por esses materiais, o inventário de massa em gramas de metais

pesados ($Z > 90$) no núcleo. Portanto, para se ter a massa em kg de metais pesados em todo o núcleo em qualquer etapa de depleção, tem-se:

```
hm_mass = operator.heavy_metal/1e3 # massa total de metais pesados no núcleo em kg
```

O operador de transporte acoplado deve estar associado a um integrador numérico, que resolverá iterativamente as equações de Bateman na forma matricial conforme mostrado na Subseção 3.3.1. O OpenMC fornece diversas classes de integradores, com uma vasta gama de graus de sofisticação algorítmica para a melhoria de convergência e precisão. Porém, novamente com o objetivo de se ter uma demanda computacional e de memória menores, o autor escolheu o método mais simples, brevemente mencionado em 3.3.1, que é o integrador baseado no método preditor. A classe associada com tal operador é denominada **PredictorIntegrator**, então deve-se instanciar um objeto da mesma.

Um integrador das equações de Bateman deve receber como argumento:

- O operador acoplado de transporte, que é o objeto operator da classe **CoupledOperator** criado acima;
- Os passos temporais de depleção. como se deseja calcular o equivalente a apenas um ano de depleção por estágio, tem-se apenas um passo temporal, com valor 1 e a unidade equivalente de um ano;
- A potência do reator, de modo a converter a unidade das taxas de reação advindas do cálculo de transporte de reações/partícula-fonte para reações/segundo de modo a inseri-las nas equações de Bateman, conforme explicado na Subseção 3.3.1.

Visto que a potência especificada para o reator é de 30 MWt, conforme mostrado na Tabela 3.4, e com base nos dados acima, o objeto integrador pode ser instanciado como:

```
power = 3e07 #watts
time_steps = [1] # 1 ano de passo temporal
integrator = openmc.deplete.PredictorIntegrator(operator, time_steps, power,
timestep_units='a')
```

No código acima, `timestep_units='a'` significa que a unidade de tempo especificado pela variável `time_steps` é anual. Por fim, de modo a iniciar a análise de depleção, dá-se um comando para que o objeto **PredictorIntegrator** inicie a integração:

```
integrator.integrate()
```

Tal linha finaliza o código em **depletionStep** correspondente ao estágio de depleção. Durante a execução, atualiza-se a composição isotópica dos materiais depletáveis do núcleo, e valores de k_{eff} para o núcleo ao início e ao fim da faixa temporal de depleção são calculados pelo código de transporte acoplado. Por fim, conforme já mencionado, a composição isotópica dos materiais de cada célula depletável são gravados em **_DEPL_MAT** para uso pelos posteriores estágios de depleção e ajuste termohidráulico.

3.4 – Referências do Capítulo

- [1] – Hashim, M.; Cao, L.; Zhou, S.; Ma, R.; Shao, Y.; Chen, R. **SPARK-NC: A Lead-Bismuth-Cooled Small Modular Fast Reactor with Natural Circulation and Load Following Capabilities**. *Energies*, Vol. 13, 5410, 21 páginas, 2020.
- [2] – Vogt, Jean-Bernard; Bouquerel, Jérémie; Carle, Carla; Proriol Serre, Ingrid. **Stability of fatigue cracks at 350 °C in air and in liquid metal in T91 martensitic steel**. *International Journal of Fatigue*, Vol. 130, 22 páginas, 2020.
- [3] – Zhou, S.; Chen, R.; Shao, Y.; Cao, L.; Bai, B.; Wu, H. **Conceptual core design study of an innovative small transportable lead-bismuth cooled fast reactor (SPARK) for remote power supply**. *International Journal of Energy Research*, 16 páginas, 2018.
- [4] – Zhou, S.; Cao, L.; Wu, H. Bai, B.; Lei, C.; Zhang, Y.. **SPARK-LS: A novel design of a compact liquid chloride salt cooled fast reactor based on tube-in-duct fuel concept**. *Annals of Nuclear Energy*, Vol. 128, 8 páginas, 2019.
- [5] – Ceder, J. **Characterization of Uranium Oxide Powders and Sinterability**. Tese, Instituto Real de Tecnologia de Estocolmo (*Kungliga Tekniska Högskolan - KTH*), 2021.
- [6] – Fazio, C. ; Sobolev, V.P.; Aerts, A.; Gavrilov, S.; Lambrinou, K.; Schuurmans, P.; Gessi, A.; Agostini, P.; Ciampichetti, A.; Martinelli, L.; Gosse, S.; Balbaud-Celerier, F.; Courouau, J.L.; Terlain, A.; Li, N.; Glasbrenner, H.; Neuhausen, J.; Heinitz, S.; Zanini, L.; Dai, Y.; Jolkkonen, M.; Kurata, Y.; Obara, T.; Thiolliere, N.; Martin-Munoz, F.J.; Heinzl, A.; Weisenburger, A.; Mueller,

G.; Schumacher, G.; Jianu, A.; Pacio, J.; Marocco, L.; Stieglitz, R.; Wetzell, T.; Daubner, M.; Litfin, K.; Vogt, J.B.; Proriol-Serre, I.; Gorse, D.; Eckert, S.; Stefani, F.; Buchenau, D.; Wondrak, T.; Hwang, I.S. **Handbook on Lead-bismuth Eutectic Alloy and Lead Properties, Materials Compatibility, Thermal-hydraulics and Technologies**. Edição 2015. OECD-Nuclear Energy Agency, Paris (França), 2016.

[7] – Lamarsh, J. R. **Introduction to Nuclear Engineering**. Addison-Wesley, 2ª Edição, 1983.

[8] – Documentação da versão 15.0 do OpenMC - docs.openmc.org – “User’s Guide - Specifying Tallies” – Acessado em 05/03/2025.

[9] – Gladstone, S.; Sesonske, A. **Nuclear Reactor Engineering**. Van Nostrand Reinhold; 3ª Edição, 1981.

[10] – Ellis, Matthew; Forget, Benoit; Smith, Kord. **Preliminary Coupling of the Monte Carlo Code OpenMC and the Multiphysics Object-Oriented Simulation Environment (MOOSE) For analyzing Doppler Feedback on Monte Carlo Simulations**. Anais do ANSMC2015, Nashville, Tennessee, EUA. 19-23 de Abril, 2015, 18 páginas.

[11] – Cheng, X.; Tak, N. **Investigation on Turbulent Heat Transfer to Lead–Bismuth Eutectic Flows in Circular Tubes for Nuclear Applications**. Nuclear Engineering and Design, 9 páginas, 2006.

[12] – Rodriguez, S.; Fathi, N.; Pourghasemi, M. **Theoretical Approach for the Fast Estimation of the Turbulent Kinematic Viscosity for Internal Flows**. ASME Journal of Nuclear Engineering and Radiation Science, 4 páginas, 2022.

[13] – Klimenko, A.V. ; Zorin, V.M. . **Thermal and Nuclear Power Plants/Handbook**. MEI Press, 2003.

[14] www.metalspiping.com. - T91 Steel. Acessado em 05/03/2025.

[15] - mooseframework.inl.gov/bison/theory/gap_plenum_models.html . Acessado em 05/03/2025.

- [16] – Calabrese, R.; Schubert, A.; Van Uffelen, P.; Vlahovic, L.; Győri, Cs. **Upper plenum temperature calculations: comparison of TRANSURANUS with a 2-D model under steady-state conditions.** Anais do *NENE2014*, Portorož, Eslovênia, 8-11, de Setembro, 2014, 8 páginas.
- [17] – Ainscough, J. B. **Gap Conductance in Zircaloy-Clad LWR Fuel Rods.** Relatório técnico para OECD/NEA, Abril de 1982.
- [18] – Kadioglu, Samet Y. ; Nourgaliev, Robert R.; Mousseau, Vincent A. **A comparative study of the harmonic and arithmetic averaging of diffusion coefficients for non-linear heat conduction problems.** Relatório técnico para o *Idaho National Laboratory* (INL), Idaho Falls, EUA, 2008.
- [19] – Iguchi, Manabu; Ilegbusi, Olusegun J. **One-dimensional Heat Conduction.** *Basic Transport Phenomena in Materials Engineering*, Springer, 2013.
- [20] – Guyer, J. E. ; Wheeler, D. ; Warren, J. A. **FiPy: Partial Differential Equations with Python.** *Computing in Science & Engineering*, 10 páginas, 2009.
- [21] – Terremoto, L. A. A. **Notas de aula para “Fundamentos de Engenharia Nuclear-Reatores”.** Instituto de Pesquisas Energéticas e Nucleares – IPEN/CNEN – São Paulo, 2004.
- [22] - Documentação da versão 15.0 do OpenMC - ***docs.openmc.org*** – “*User’s Guide – Depletion and Transmutation*” – Acessado em 05/03/2025.
- [23] - Documentação da versão 15.0 do OpenMC - ***docs.openmc.org*** – “*Examples – Depletion*” – Acessado em 05/03/2025.
- [24] – ***www.openmc.org*** – “*Data – Depletion Chains*” – Acessado em 05/03/2025.
- [25] - Kim, Kang Seog. **Specification for the VERA Benchmark Suite.** Relatório Técnico CASL-U-2015-1014-000, Revisão 0, 2016.

CAPÍTULO 4

RESULTADOS E DISCUSSÃO

Este capítulo mostra os resultados obtidos ao se utilizar o código de depleção com ajuste termohidráulico para realizar a comparação com as curvas e parâmetros do SPARK-NC mostrados em [1], sendo essa comparação o objetivo subsidiário mencionado na Seção 1.4 que valida os objetivos principais da dissertação, isto é, a modelagem do núcleo do SPARK-NC e a metodologia de acoplamento neutrônico/termohidráulico descritos no Capítulo 3.

A execução da simulação completa, isto é, da análise de depleção entre $t = 0$ e $t = 8.4$ anos, foi feita em uma máquina com processador Intel i9[®] de 32 núcleos, e cada estágio de ajuste termohidráulico foi feito com 1200 bateladas de 35000 partículas cada, com 70 bateladas inativas e interrupções para atualização da temperatura a cada 100 bateladas, enquanto cada estágio de depleção foi feito com 600 bateladas também com 35000 partículas e 70 bateladas inativas (o que também resulta em 1200 bateladas, pois a depleção entre o intervalo $t = [N, N + 1]$ anos faz dois cálculos de criticalidade, um para $t = N$ e outro para $t = N + 1$). Para tal configuração, a simulação durou um tempo total de aproximadamente 60 horas, com o núcleo modelado em sua totalidade, sem a aplicação de simetria com condições de contorno periódicas (tal simplificação poderia ser aplicada pois o núcleo possui simetria de rotação de 120°). Conforme mencionado na Tabela 3.5, a biblioteca de seções de choque utilizada na simulação foi a ENDF-VII.0 para fins de consistência, pois é aquela utilizada pelo SARAX para a obtenção dos resultados em [1].

As próximas subseções dedicam-se:

- Aos resultados relacionados à depleção, ao se comparar a curva de depleção do artigo [1] com aquela obtida ao longo da simulação, além das diferenças de massa de ^{235}U , ^{238}U e ^{239}Pu apontadas em [1] com aquelas também calculadas na simulação;
- Aos resultados relacionados ao ajuste termohidráulico, onde as curvas obtidas ao longo da simulação para os perfis axiais de temperatura do centro do cerne combustível, do revestimento e do fluido refrigerante (LBE) da vareta mais quente (ou seja, a vareta representativa do núcleo) são comparados com o valor mostrado no artigo [1] para esses perfis. Adicionalmente, são mostrados mapas transversais e isométricos de distribuição de temperatura no núcleo;
- Ao resultado relacionado à curva de perfil de potência axial relativa do núcleo, comparando-a com aquela encontrada em [1].

4.1 – Resultados de depleção

Conforme mencionado na Subseção 3.3.2, foi utilizado para a simulação um arquivo de cadeia de depleção simplificado para reatores de espectro rápido. Para fins informativos, os isótopos constituintes da cadeia são mostrados na Tabela 4.1.

Tabela 4.1 – Isótopos constituintes da cadeia de depleção utilizada.

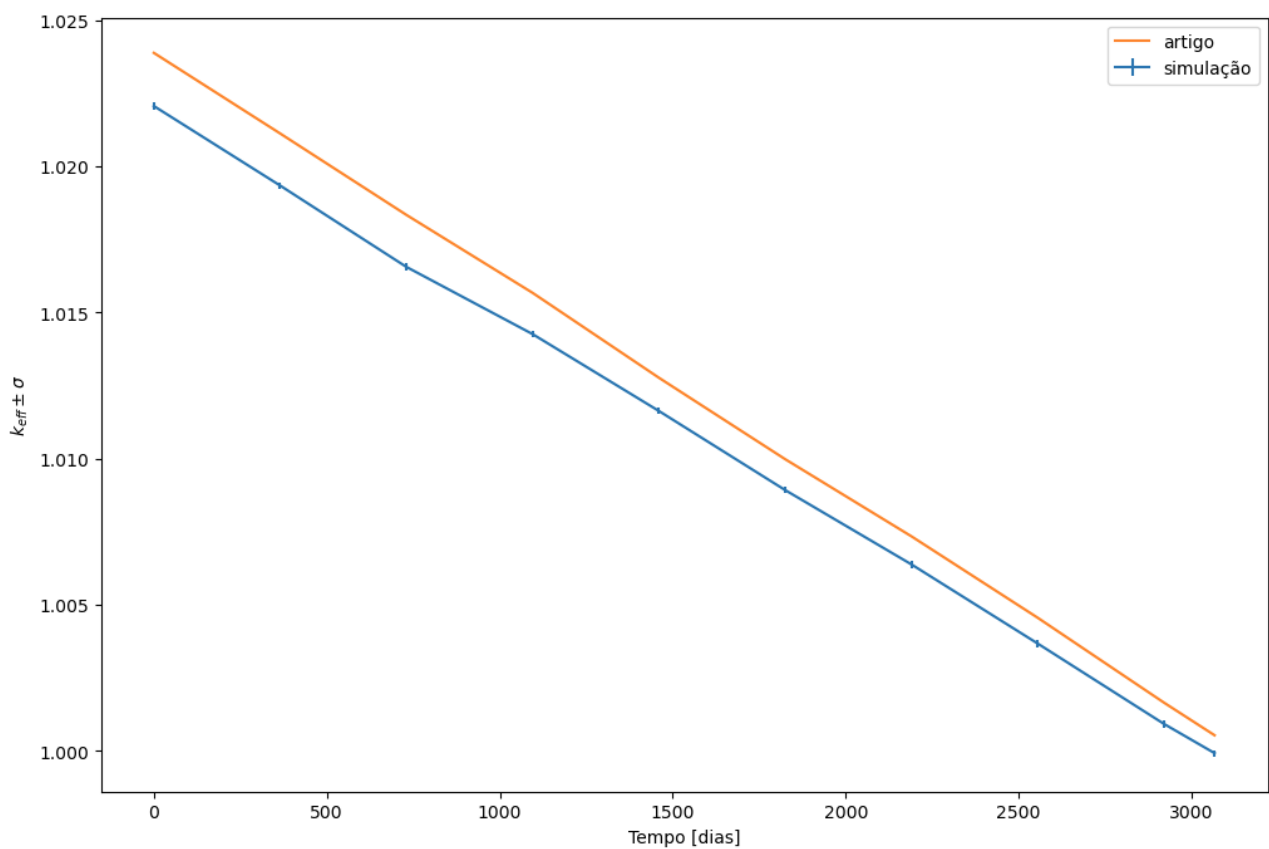
^{10}B , ^{11}B	^{16}O	^{81}Br , ^{82}Br	^{82}Kr , ^{83}Kr , ^{84}Kr , ^{84}Kr , ^{85}Kr , ^{86}Kr	^{89}Sr , ^{90}Sr
^{89}Y , ^{90}Y , ^{91}Y	^{90}Zr , ^{91}Zr , ^{92}Zr , ^{93}Zr , ^{94}Zr , ^{95}Zr , ^{96}Zr	^{95}Nb	^{92}Mo , ^{94}Mo , ^{95}Mo , ^{96}Mo , ^{97}Mo , ^{98}Mo , ^{99}Mo , ^{100}Mo	^{99}Tc , $^{99\text{m}}\text{Tc}$, ^{100}Tc
^{100}Ru , ^{101}Ru , ^{102}Ru , ^{103}Ru , ^{104}Ru , ^{105}Ru , ^{106}Ru	^{102}Rh , $^{102\text{m}}\text{Rh}$, ^{103}Rh , $^{103\text{m}}\text{Rh}$, ^{104}Rh , ^{105}Rh , $^{105\text{m}}\text{Rh}$, ^{106}Rh , $^{106\text{m}}\text{Rh}$	^{104}Pd , ^{105}Pd , ^{106}Pd , ^{107}Pd , ^{108}Pd , ^{109}Pd	^{107}Ag , ^{109}Ag , $^{109\text{m}}\text{Ag}$, ^{110}Ag , $^{110\text{m}}\text{Ag}$, ^{111}Ag	^{110}Cd , ^{111}Cd , ^{112}Cd , ^{113}Cd , ^{114}Cd , ^{115}Cd ,
^{113}In , ^{115}In	^{121}Sb , ^{122}Sb , ^{125}Sb , ^{127}Sb	^{127}Te , $^{127\text{m}}\text{Te}$, $^{129\text{m}}\text{Te}$, ^{132}Te	^{127}I , ^{128}I , ^{130}I , ^{131}I , ^{132}I , ^{135}I	^{128}Xe , ^{130}Xe , ^{131}Xe , ^{132}Xe , ^{133}Xe , ^{134}Xe , ^{135}Xe , $^{135\text{m}}\text{Xe}$, ^{136}Xe , ^{137}Xe
^{133}Cs , ^{134}Cs , ^{135}Cs , ^{136}Cs , ^{137}Cs	^{134}Ba , ^{137}Ba , ^{140}Ba	^{139}La , ^{140}La	^{140}Ce , ^{141}Ce , ^{143}Ce , ^{144}Ce	^{141}Pr , ^{142}Pr , ^{143}Pr , ^{144}Pr
^{142}Nd , ^{143}Nd , ^{144}Nd , ^{145}Nd , ^{156}Nd , ^{147}Nd , ^{148}Nd , ^{149}Nd , ^{150}Nd , ^{151}Nd	^{147}Pm , ^{148}Pm , $^{148\text{m}}\text{Pm}$, ^{149}Pm , ^{150}Pm , ^{151}Pm	^{159}Tb , ^{160}Tb , ^{161}Tb	^{147}Sm , ^{148}Sm , ^{149}Sm , ^{150}Sm , ^{151}Sm , ^{152}Sm , ^{153}Sm , ^{154}Sm , ^{155}Sm	^{151}Eu , ^{152}Eu , $^{152\text{m}}\text{Eu}$, ^{153}Eu , ^{154}Eu , ^{155}Eu , ^{156}Eu , ^{157}Eu
^{152}Gd , ^{154}Gd , ^{155}Gd , ^{156}Gd , ^{157}Gd , ^{158}Gd , ^{159}Gd , ^{160}Gd , ^{161}Gd	^{160}Dy , ^{161}Dy , ^{162}Dy , ^{163}Dy , ^{164}Dy , ^{165}Dy	^{165}Ho	^{162}Er , ^{164}Er , ^{166}Er , ^{167}Er , ^{168}Er , ^{169}Er , ^{170}Er , ^{171}Er	^{169}Tm , ^{170}Tm , ^{171}Tm
^{174}Hf , ^{175}Hf , ^{177}Hf , ^{178}Hf , ^{179}Hf , ^{180}Hf , ^{181}Hf	^{230}Th , ^{231}Th , ^{232}Th , ^{233}Th , ^{234}Th	^{181}Ta , ^{182}Ta	^{231}Pa , ^{232}Pa , ^{233}Pa , ^{234}Pa	^{232}U , ^{233}U , ^{234}U , ^{235}U , ^{236}U , ^{237}U , ^{238}U , ^{239}U
^{236}Np , ^{237}Np , ^{238}Np , ^{239}Np , ^{240}Np , $^{240\text{m}}\text{Np}$	^{236}Pu , ^{237}Pu , ^{238}Pu , ^{239}Pu , ^{240}Pu , ^{241}Pu , ^{242}Pu , ^{243}Pu	^{241}Am , ^{242}Am , $^{242\text{m}}\text{Am}$, ^{243}Am , ^{244}Am , $^{244\text{m}}\text{Am}$	^{242}Cm , ^{243}Cm , ^{244}Cm , ^{245}Cm , ^{246}Cm	

A comparação entre os valores de k_{eff} do artigo [1] (k_{eff}^A - ver Figura 3.1) e aqueles obtidos pela simulação (k_{eff}^S) são mostrados na Figura 4.1. A Tabela 4.2 mostra o valor de k_{eff} encontrado em cada ano, juntamente com a diferença entre k_{eff}^A e k_{eff}^S dada em *pcm* (*per cent mille*), isto é, $10^5 \times |k_{eff}^A - k_{eff}^S|$. O desvio padrão, para todos os anos, ficou em torno de $\sigma = 0.00011$ (11 pcm).

Tabela 4.2 – Comparação entre os valores de k_{eff} da simulação (k_{eff}^S) e do artigo [1] (k_{eff}^A), com a diferença em pcm.

t	k_{eff}^A	k_{eff}^S	Diferença em pcm
0	1.02388	1.02206	182
1	1.02112	1.01934	178
2	1.01833	1.01656	177
3	1.01567	1.01426	141
4	1.01276	1.01162	114
5	1.00998	1.00892	106
6	1.0073	1.00638	92
7	1.00455	1.00367	88
8	1.00165	1.00092	73
8.4	1.00053	0.99992	61

Figura 4.1 – Comparação da evolução temporal do valor de k_{eff} durante a depleção entre $t = 0$ e $t = 8,4$ anos. Os valores em laranja correspondem aos valores do artigo original [1], enquanto os valores em azul correspondem à simulação. Os traços verticais na curva de simulação correspondem à faixa do desvio padrão.



A Figura 4.2 mostra o erro relativo percentual, dado por

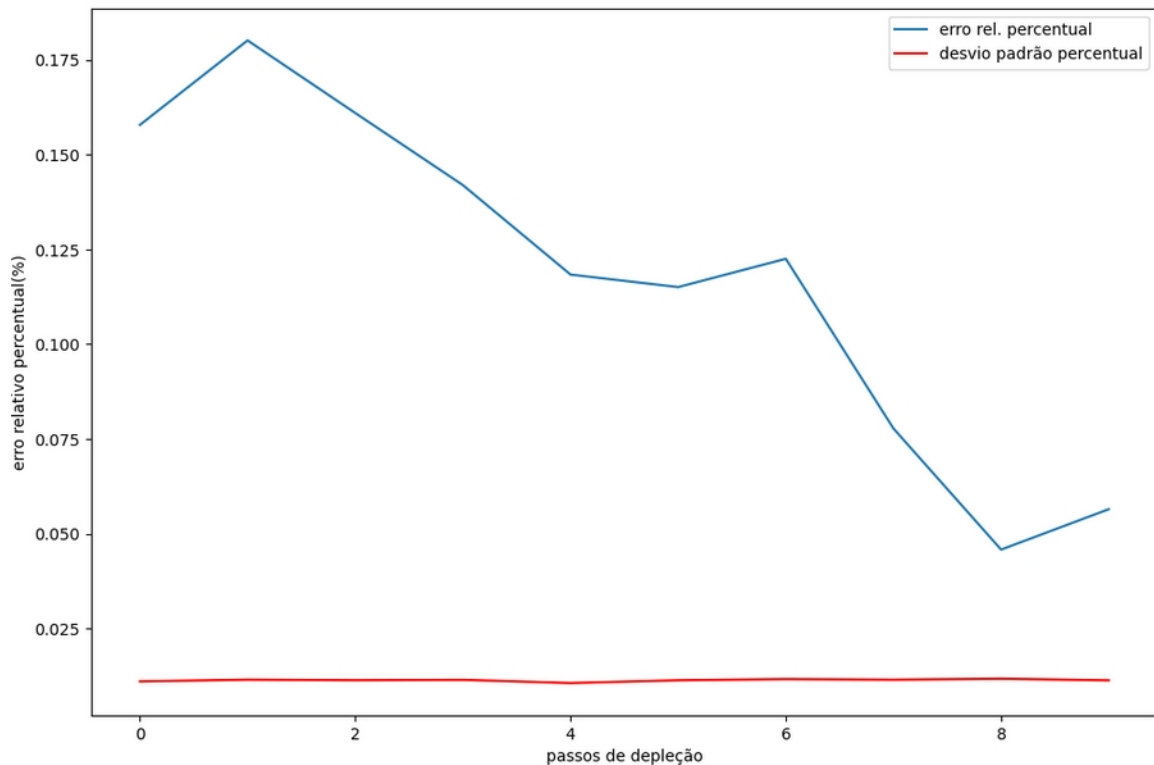
$$E_{\%} = 100 \times \left(\frac{|k_{eff}^A - k_{eff}^S|}{k_{eff}^A} \right),$$

e o desvio padrão percentual, dado por

$$\sigma_{\%} = 100 \times \left(\frac{\sigma}{k_{eff}^S} \right),$$

onde k_{eff}^A é o valor de k_{eff} do artigo e k_{eff}^S é o valor de k_{eff} obtido na simulação. Nota-se que, apesar do erro relativo percentual estar acima do desvio padrão percentual em todos os anos, o valor desse erro é relativamente reduzido em toda a simulação, alcançando um máximo de 0.18% em $t = 1$ ano e uma média de 0.11%. Nota-se também que nos últimos anos da simulação o erro percentual tende a diminuir, mas tal comportamento se deve à diferença de inclinação entre a curva simulada e a curva do artigo, o que por sua vez deve-se provavelmente à propagação de erros na composição isotópica ao longo do cálculo de depleção.

Figura 4.2 – Erro relativo percentual e desvio padrão percentual ao longo da simulação.



Os resultados da variação de massa de ^{235}U , ^{238}U e ^{239}Pu entre $t=0$ e $t=8$ anos na simulação são comparados com aqueles mostrados no artigo na Tabela 4.3. Nota-se que o erro percentual também é relativamente reduzido, com média de 0.79% e tendo seu maior valor para o ^{235}U , com 1.74%.

Tabela 4.3 – Comparação entre os valores do artigo original [1] e da simulação para a diferença de massa de ^{235}U , ^{238}U e ^{239}Pu entre $t=0$ e $t=8$ anos.

Isótopo	Valor Simulação (kg)	Valor Artigo (kg)	Erro percentual (%)
^{235}U	105.63	107.5	1.74
^{238}U	63.84	63.1	0.11
^{239}Pu	-51.68	-51.4	0.54

4.2 – Resultados do ajuste termohidráulico

Esta subseção discute alguns resultados referentes aos estágios de ajuste termohidráulico. Inicialmente, são mostrados dados que corroboram as hipóteses discutidas na Subseção 3.2.1. Em primeiro lugar, discute-se sobre a hipótese em que os perfis axiais de densidade linear de potência $q_v(z)$ de cada vareta combustível v tem o mesmo formato do perfil axial médio $\bar{q}(z)$, com a variação apenas da magnitude entre as varetas. A Figura 4.3 mostra a distribuição entre as varetas combustíveis do desvio percentual de formato durante o primeiro ajuste termohidráulico, definido da seguinte forma: toma-se um determinado número n_a de pontos amostrais entre (z_{min}, z_{max}) correspondentes às alturas mínimas e máximas do cerne da vareta e, com base nesses pontos normaliza-se $q_v(z)$ e $\bar{q}(z)$ no intervalo $(0, 1)$, para que não haja diferença de magnitude dos perfis e apenas o formato seja analisado, obtendo-se então $q_v^n(z)$ e $\bar{q}^n(z)$:

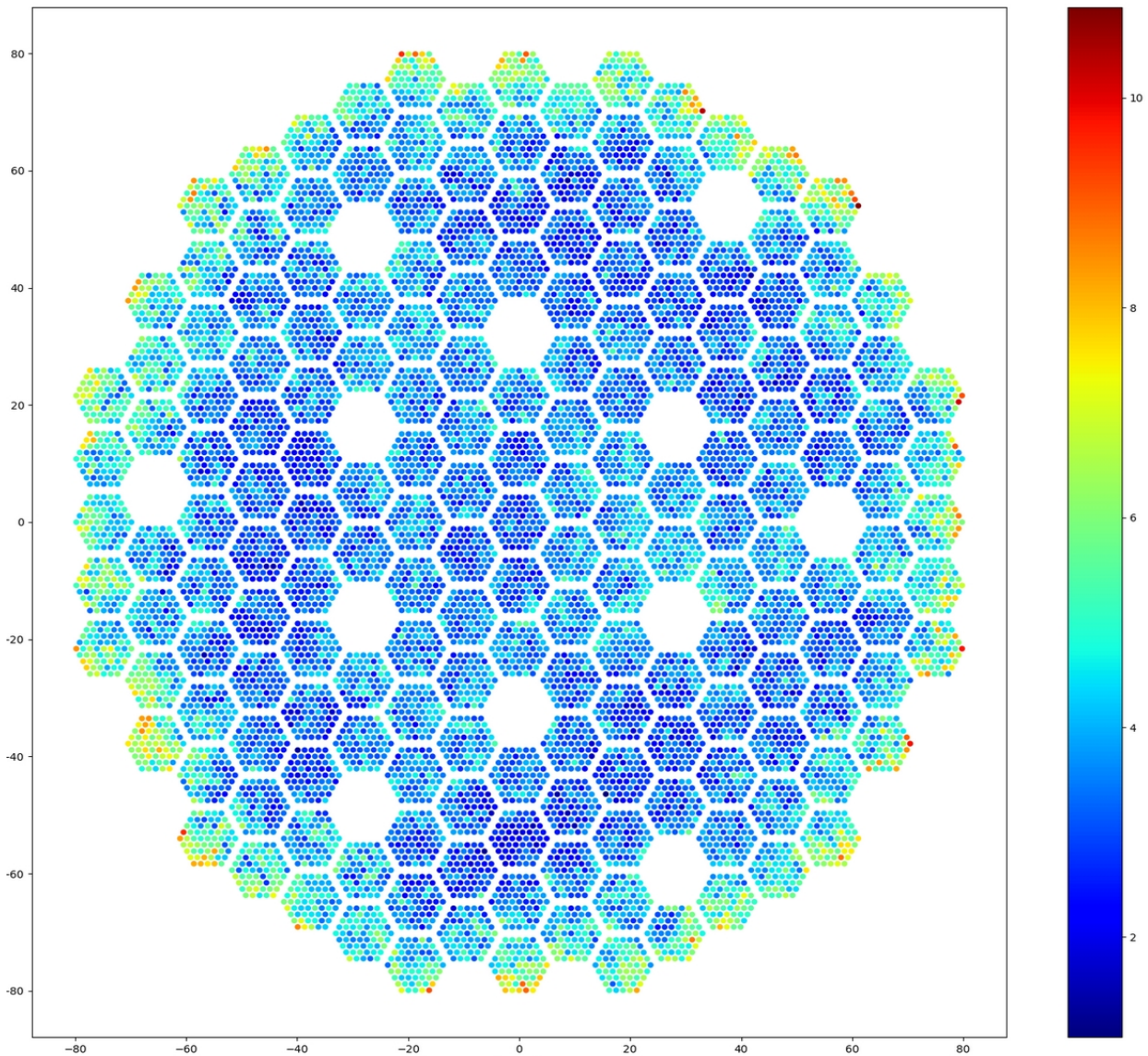
$$q_v^n(z_a) = \frac{q_v(z_a)}{\max_{z_a} q_v(z)}, \quad \bar{q}^n(z_a) = \frac{\bar{q}(z_a)}{\max_{z_a} \bar{q}(z)},$$

onde z_a denota qualquer um dos n_a pontos amostrados entre (z_{min}, z_{max}) . Para o cálculo do desvio percentual em questão, tomou-se $n_a = 30$ pontos. O desvio percentual de formato de $q_v(z)$ da vareta v em relação ao perfil axial de densidade de potência médio $\bar{q}(z)$ é então dado por:

$$\Delta_q^v = 100 \times \frac{1}{n_a} \sum_{z_a} \left| \frac{q_v^n(z_a) - \bar{q}^n(z_a)}{\bar{q}^n(z_a)} \right|.$$

Observa-se pela Figura 4.3 que para grande parte das varetas combustíveis, o desvio percentual de formato está em cerca de 3%, e o desvio máximo é de cerca de 10.5%. É notado que os maiores percentuais de desvio ocorreram na fronteira da região das varetas combustíveis conforme esperado, pois é a região mais próxima aos elementos refletor e os elementos absorvedores das varetas de blindagem, que tendem a deformar o perfil axial de fluxo. A média dos desvios percentuais dentre todas as varetas combustíveis é de aproximadamente 3.8%, o que mostra que a hipótese de isotropia do formato do perfil axial médio de potência é razoável.

Figura 4.3 – Distribuição dentre as varetas combustíveis do desvio percentual de formato em relação ao perfil axial médio de densidade de potência $\bar{q}(z)$. A média dentre todos os desvios é de ~3.8%, indicando um bom nível de isotropia de formato dentre as varetas.



Também é analisada a hipótese de que a geração de calor dá-se de maneira largamente majoritária no cerne das varetas, e portanto pode-se considerar que $q(r, \theta, z)$, a densidade volumétrica de produção calorífica, pode ser considerada como sendo nula fora da região das varetas. De fato, ao se tomar o *tally* médio, considerando-se todos os estágios da simulação, da soma das produções caloríficas Q_v nos cernes de todas as varetas combustíveis e multiplicar pelo fator f que converte da unidade de produção calorífica normalizada por partícula-fonte [eV/pf] para a potência [W], tem-se:

$$f \left(\sum_v \bar{Q}_v \right) \approx 2.97 \times 10^7 W,$$

que é um valor muito próximo da potência total do núcleo, que é de $30 \text{ MW} = 3 \times 10^7 \text{ W}$ térmicos. Portanto, tal cálculo mostra que esta também é uma hipótese realista, pois, na média, 99.3% da potência é gerada na região do cerne das varetas combustíveis durante o ciclo de vida do reator.

A Figura 4.6 mostra o perfil empregado como a distribuição de densidade volumétrica de potência da vareta representativa $\dot{q}_R(z)$. A Figura 4.7 mostra a distribuição de temperatura no domínio da Figura 3.16 para a vareta representativa, $T^R(\vec{r})$ obtida ao final do estágio de ajuste termohidráulico correspondente ao início do ciclo de vida do reator, enquanto a Figura 4.8 compara as distribuições axiais de temperatura da vareta representativa na região do cerne ($T^R(z)$, $30 \text{ cm} \leq z \leq 120 \text{ cm}$), obtidas pela simulação numérica também ao final do estágio de ajuste termohidráulico correspondente ao início do ciclo de vida do reator, com os valores obtidos no artigo (ver Figura 8 em [1]) ao longo de 3 locais: No eixo que passa pelo centro da vareta, ao longo do revestimento (eixo tomado à metade da espessura da região do revestimento no domínio do modelo) e no fluido refrigerante (tomado também à metade da espessura da região de fluido refrigerante no domínio do modelo).

Figura 4.6 – Perfil axial de densidade volumétrica de potência da vareta representativa $\dot{q}_R(z) = f\beta_R[Q(z)/A]$ tomado na região do cerne da vareta ($30 \text{ cm} \leq z \leq 120 \text{ cm}$), ao final do estágio de ajuste termohidráulico correspondente ao início do ciclo de vida do reator.

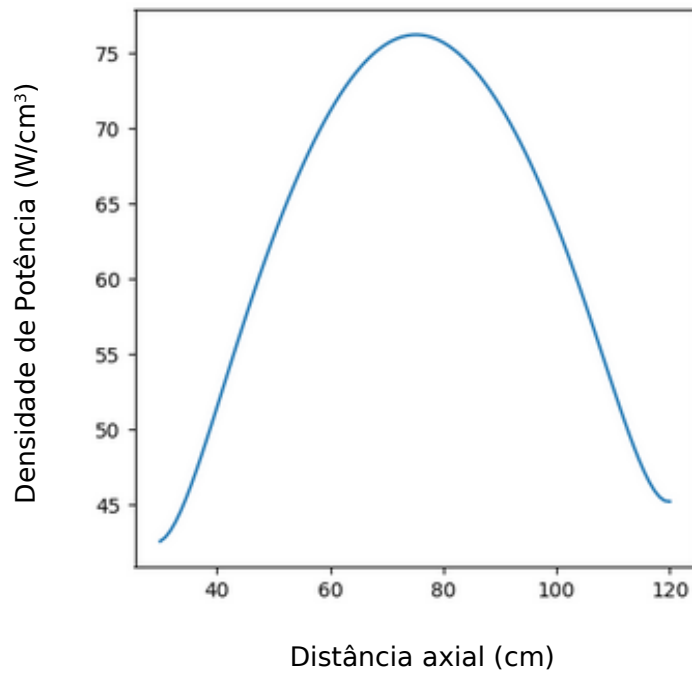


Figura 4.7 – Distribuição de temperatura da vareta representativa, $T^R(\vec{r})$, no domínio axissimétrico da Figura 3.16, ao final do estágio de ajuste termohidráulico correspondente ao início do ciclo de vida do reator. Os eixos r e z estão em cm, e a temperatura é dada em $^{\circ}\text{C}$.

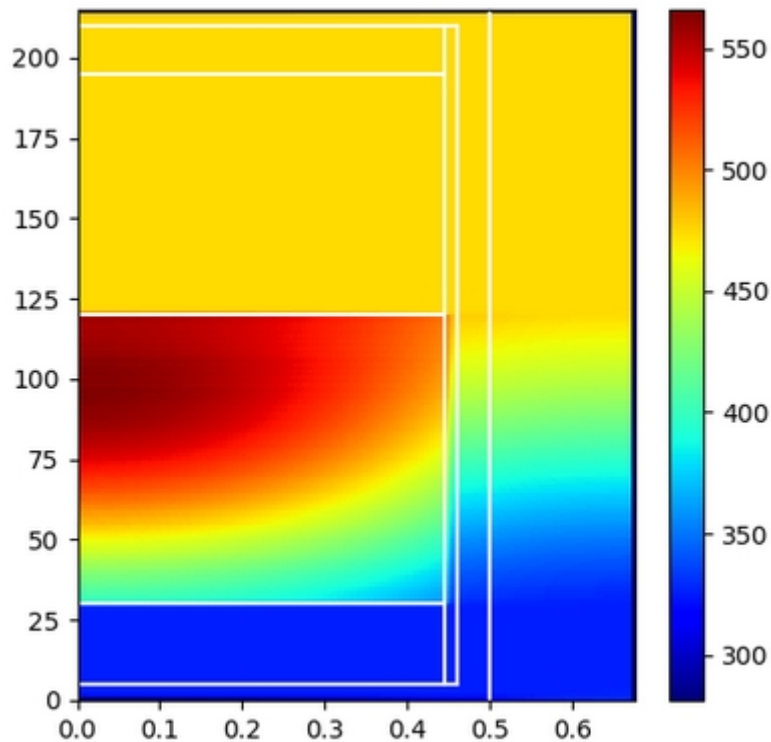
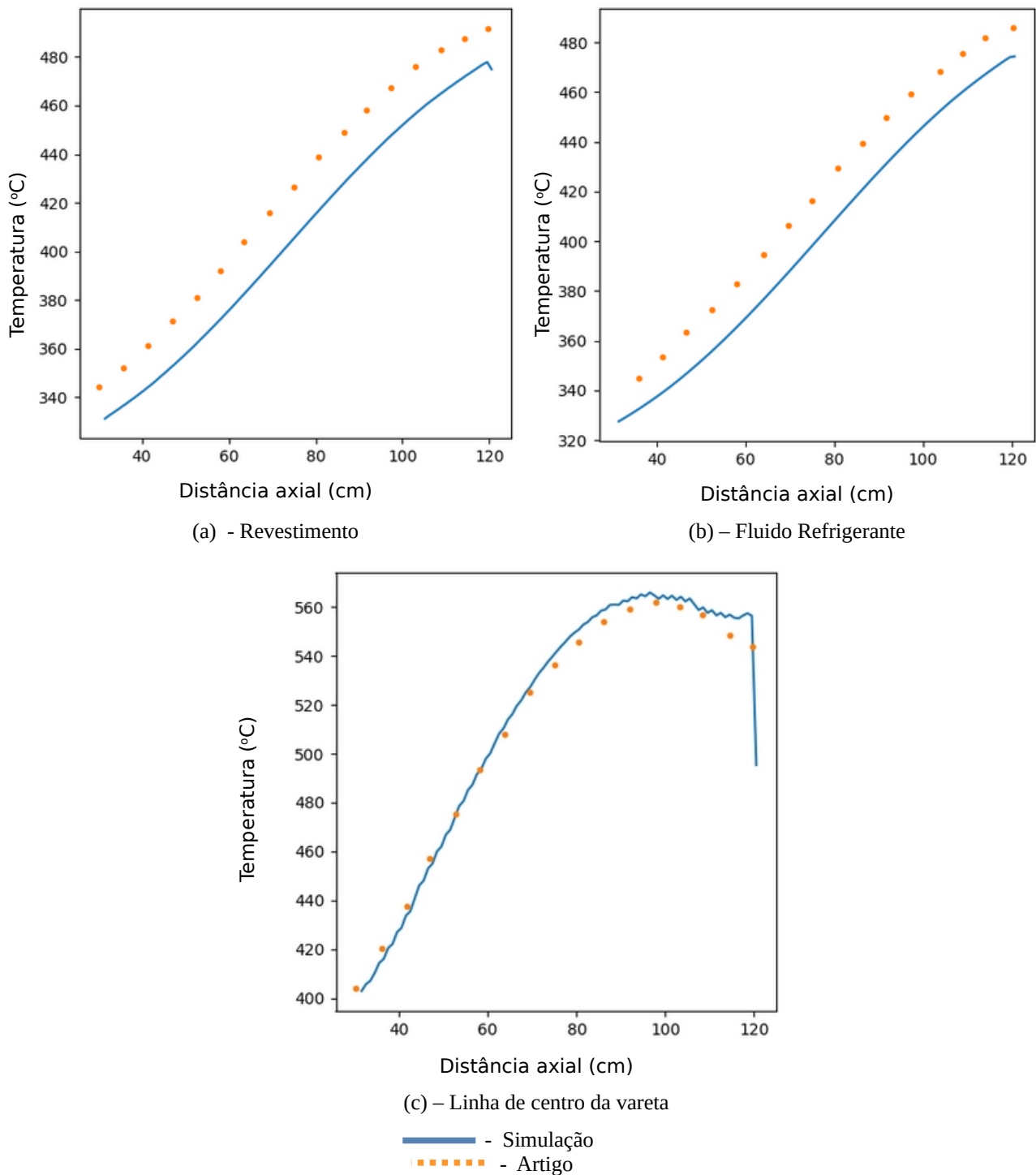


Figura 4.8 – Perfil axial de temperatura da vareta representativa na região do cerne da vareta ($T^R(z)$, $30 \text{ cm} \leq z \leq 120 \text{ cm}$), comparando os valores do artigo e da simulação, para o revestimento (a), fluido refrigerante (b) e linha de centro da vareta (c) ao final do estágio de ajuste termohidráulico correspondente ao início do ciclo de vida do reator.



A comparação entre os valores do artigo original [1] e da simulação para as distribuições de temperatura na vareta representativa mostra uma concordância satisfatória com os valores obtidos em [1] mormente no que se refere na temperatura do eixo central da vareta combustível, mostrando que o ajuste termohidráulico proposto consegue capturar com certa precisão as características

térmicas do núcleo. A média dos erros percentuais para o centro da vareta, o revestimento e o LBE são, respectivamente, 0.66%, 4.12% e 3.68%. A Figura 4.9 mostra a distribuição média, tomada na direção axial, da temperatura em todo o núcleo ao final do estágio de ajuste termohidráulico correspondente ao início do ciclo de vida do reator, enquanto a Figura 4.10 mostra uma vista isométrica de um corte longitudinal do núcleo do SPARK-NC, com o mapa de temperaturas também ao final desse estágio.

Figura 4.9 – Distribuição de temperatura média, tomada na direção axial, em todo o núcleo ao final do estágio de ajuste termohidráulico correspondente ao início do ciclo de vida do reator. As temperaturas estão em Kelvin, e a escala de cores é não-linear (sendo mais dilatada próximo às temperaturas mais altas), de modo a mostrar a variação sutil de temperatura na região mais central do núcleo .

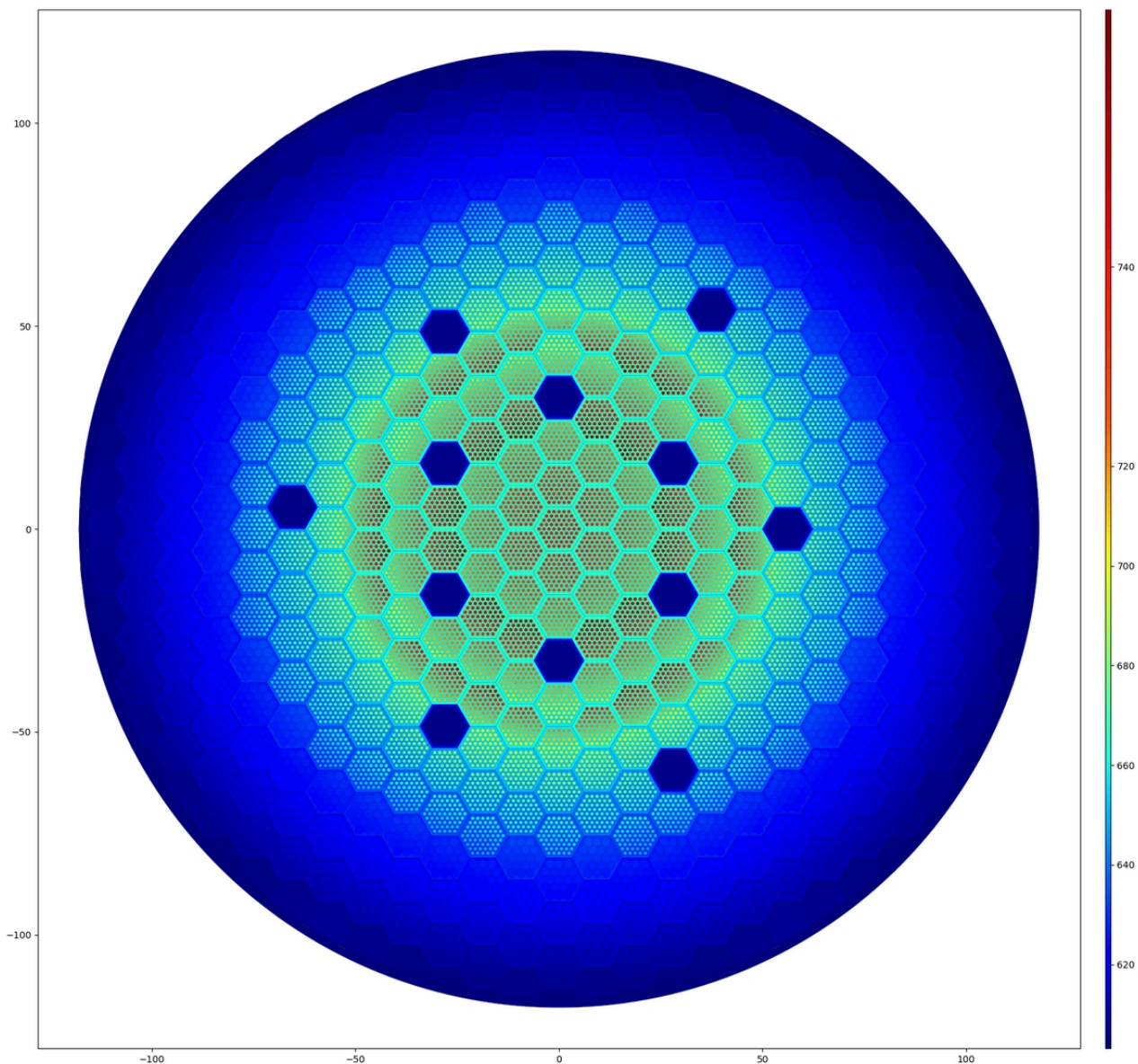
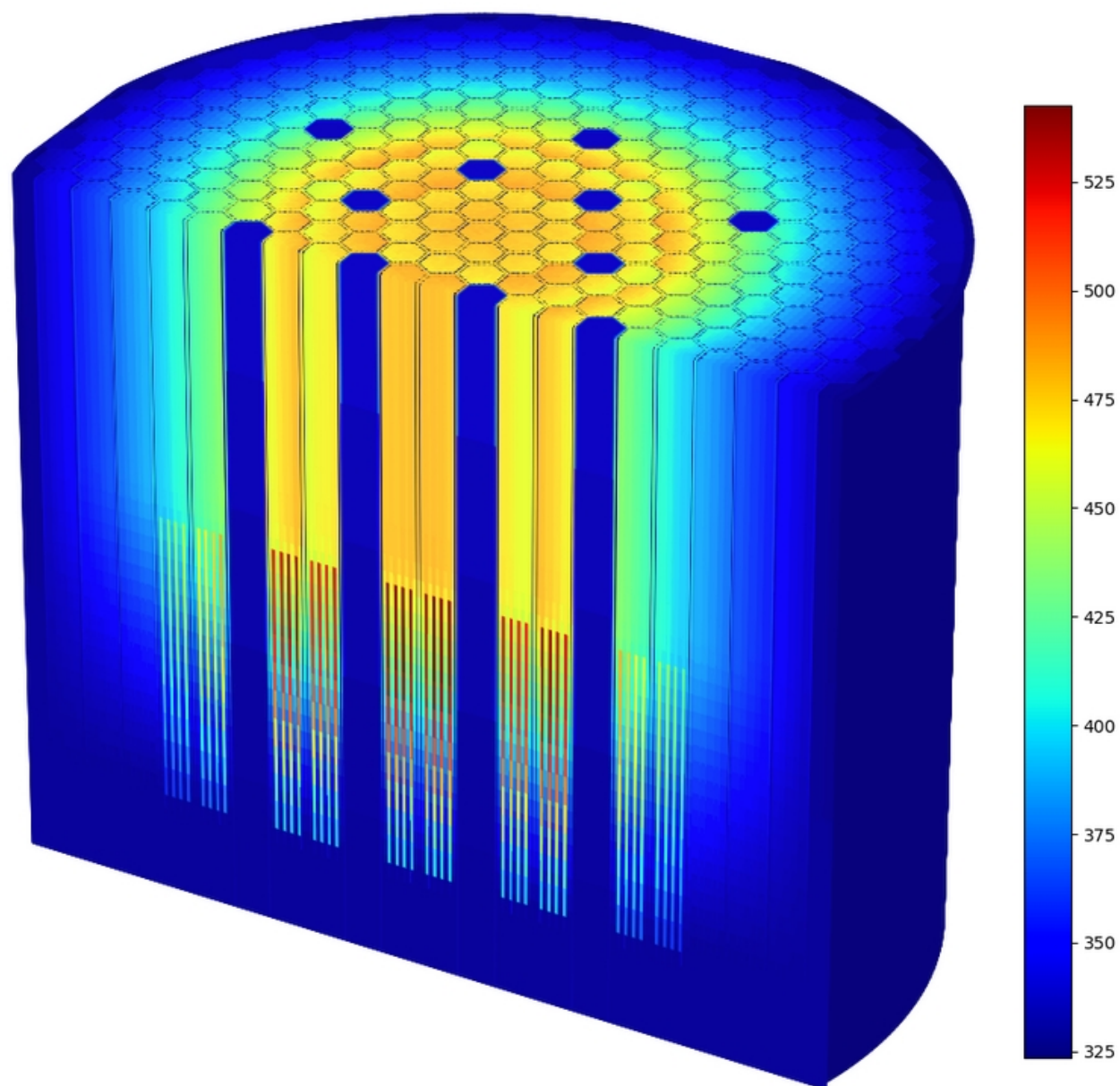


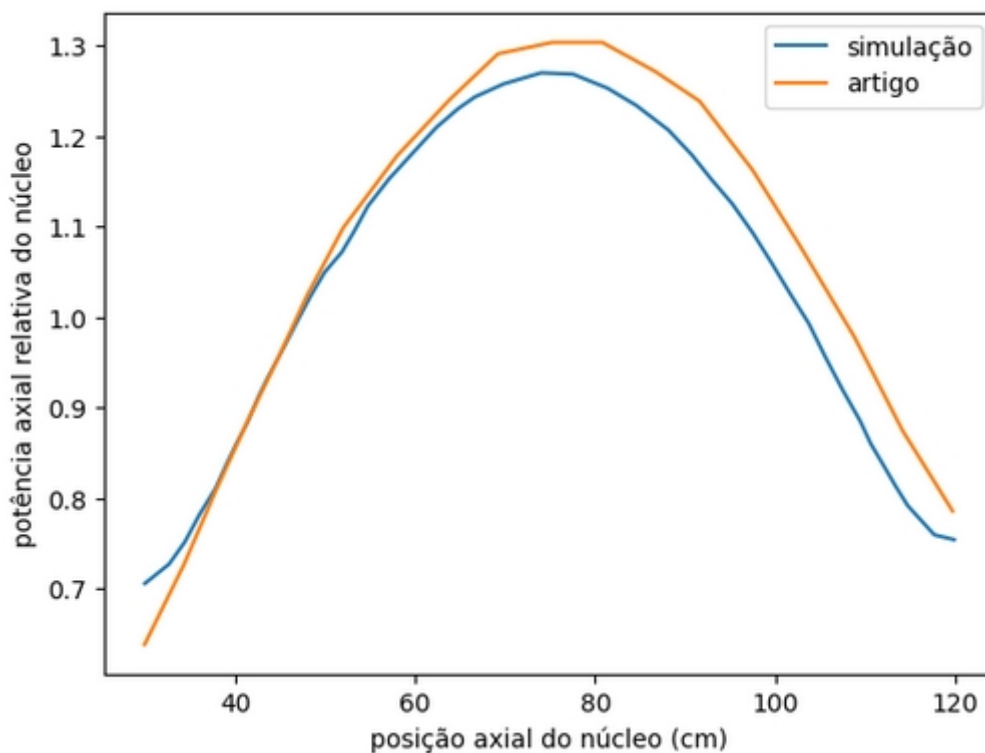
Figura 4.10 – Vista isométrica de um corte longitudinal do núcleo do SPARK-NC, mostrando o mapa de temperaturas ao final do estágio de ajuste termohidráulico correspondente ao início do ciclo de vida do reator. A escala de temperatura está em graus Celsius.



4.3 – Resultado do perfil axial de potência relativa

A Figura 4.11 compara o perfil axial de potência relativa na faixa de altura do cerne das varetas combustíveis ($30\text{ cm} \leq z \leq 120\text{ cm}$), no início do ciclo de vida do reator, obtida na simulação, com aquela encontrada em [1]. Tal perfil é calculado como a razão entre a distribuição de potência axial total gerada no núcleo e o valor médio dessa distribuição. Observa-se que as curvas possuem uma pequena diferença de magnitude no pico (1.28 na simulação e 1.31 no artigo), e com o mesmo formato senoidal geral, mas com diferenças nas extremidades, principalmente na parte inferior da região do cerne ($z = 30\text{ cm}$), o que o autor acredita que possa ser devido à diferenças de modelagem dos refletores/*plenum* entre esta dissertação e [1]. O erro percentual relativo médio para o perfil é de 3.8%.

Figura 4.11 – Comparação da potência axial relativa no início do ciclo de vida do reator obtido na simulação com aquele mostrado em [1]. O erro percentual relativo médio do perfil é de 3.52%.



4.3 – Referências do Capítulo

[1] – Hashim, M.; Cao, L.; Zhou, S.; Ma, R.; Shao, Y.; Chen, R. **SPARK-NC: A Lead-Bismuth-Cooled Small Modular Fast Reactor with Natural Circulation and Load Following Capabilities**. *Energies*, Vol. 13, 5410, 21 páginas, 2020.

CAPÍTULO 5

CONCLUSÃO E DIRECIONAMENTOS FUTUROS

Este capítulo conclui o trabalho apresentado nesta dissertação, oferecendo um resumo do que foi feito, além de discutir acerca do cumprimento dos objetivos e das dificuldades e limitações encontradas no transcurso de seu desenvolvimento. Adicionalmente, são também sugeridos direcionamentos para possíveis trabalhos futuros e dadas algumas considerações finais.

5.1 – Resumo do trabalho

Esta dissertação realizou a modelagem neutrônica e termohidráulica de um reator rápido refrigerado à liga de chumbo-bismuto eutético (LBE), denominado SPARK-NC e descrito em [1], utilizando as bibliotecas *Python* OpenMC, de cálculo de transporte de partículas baseada no método Monte Carlo, e FiPy, que soluciona equações diferenciais parciais pelo método de volumes finitos. O autor construiu um modelo do núcleo do SPARK-NC com base nos dados geométricos e de composição material, fornecidos em [1] com um elevado grau de detalhe. A partir de tal modelo, o trabalho dedicou-se à reproduzir, para fins comparativos, alguns resultados encontrados em [1]: a evolução temporal do fator de multiplicação (k_{eff}) e a diferença de massa dos isótopos ^{235}U , ^{238}U e ^{239}Pu durante o ciclo de vida do reator, os perfis axiais de temperatura em alguns pontos da vareta mais quente no início do ciclo de vida do reator e o perfil axial de potência relativa no núcleo.

Para realizar o acoplamento neutrônico/termohidráulico, calcula-se, através do FiPy, a distribuição de temperatura no domínio espacial axissimétrico de uma única vareta combustível representativa no núcleo, tendo a o perfil axial de densidade de potência dessa vareta como entrada do problema, e posteriormente mapeia-se tal distribuição para todas as outras varetas combustíveis, sendo esse mapa baseado na razão entre as potências caloríficas geradas nas outras varetas e aquela gerada na vareta representativa. O autor propôs uma metodologia, julgada inédita, em que tanto o perfil axial de densidade de potência da vareta representativa quanto os fatores que caracterizam a razão entre potências do mapa supramencionado podem ser obtidos por meio de *tallies* dados por expansões funcionais contínuas em todo domínio espacial ocupado pelas varetas combustíveis, sendo essas expansões obtidas por funções *built-in* do OpenMC. As temperaturas em outros pontos do núcleo (camada refletora, de blindagem, paredes do vaso do reator) são determinadas através de um modelo condutivo simples que toma como entrada a temperatura dos elementos combustíveis mais externos da região ativa do núcleo.

A evolução temporal do k_{eff} e a diferença de massas de isótopos durante o ciclo de vida do reator foi calculada através de uma simulação em que estágios de ajuste termohidráulico e de depleção se intercalam, fazendo com que o k_{eff} obtido em um determinado ao fim de um estágio de ajuste termohidráulico imediatamente anterior à um estágio de depleção é o k_{eff} associado à um núcleo cuja distribuição de temperatura em estado estacionário é induzida pelo fluxo neutrônico em estado estacionário que corresponde à configuração isotópica do núcleo para aquele estágio de depleção.

O estágio termohidráulico consiste de uma simulação para a estimação de k_{eff} que é interrompida ao longo de pontos predefinidos, sendo cada interrupção caracterizada pelo processo de leitura de *tallies*, cálculo da distribuição de temperatura da vareta representativa e atribuição da distribuição de temperaturas no núcleo descrita acima, quando a simulação retorna então à estimação de k_{eff} com as temperaturas do núcleo atualizadas. Tais interrupções têm o intuito de modificar iterativamente a distribuição de temperatura do núcleo até que a mesma convirja para o que se espera ser a distribuição de temperatura em estado estacionário para aquela configuração do núcleo.

O estágio de depleção simula a alteração na configuração isotópica do núcleo devido à queima de seu combustível no decurso de uma determinada faixa temporal. Para alcançar tal objetivo, utiliza-se o módulo de depleção do OpenMC, que toma como entrada um arquivo de dados de depleção (*depletion chain file*, ou arquivo de cadeia de depleção). Neste trabalho, a faixa temporal correspondente à cada estágio é de um ano, exceto pelo último estágio, que é de 0,4 ano, com um tempo de depleção total de 8.4 anos, pois é o tempo total especificado em [1].

A comparação entre os resultados obtidos para o modelo do SPARK-NC que foi objeto desta dissertação, e aqueles encontrados em [1] foram considerados como satisfatórios, dadas as limitações impostas pela falta de certos dados sobre a composição material do núcleo, completa ausência de informações acerca do modelo e condições de contorno utilizados para as simulações neutrônicas e termohidráulicas e diferenças inerentes entre os códigos empregados pelo autor da dissertação e no artigo [1] (ver Seções 5.2 e 5.3). Não obstante tais dificuldades, a análise desses resultados permite concluir que o modelo proposto neste trabalho foi capaz de capturar o comportamento do núcleo do SPARK-NC observado em [1] com um nível razoável de precisão.

5.2 – Cumprimento dos objetivos

No Capítulo 1, foram elencados os objetivos estipulados para a dissertação. O primeiro objetivo diz respeito à modelagem do reator SPARK-NC com a utilização da biblioteca *Python* OpenMC, utilizando os dados geométricos e de composição do mesmo, conforme fornecido em [1].

Considera-se que tal objetivo foi cumprido, pois o modelo geométrico do núcleo foi reproduzido de maneira precisa (ver Figuras 3.3 e 3.11/3.12), e os materiais foram reproduzidos dentro dos limites das informações disponíveis em [1] (ver limitações na Seção 5.3).

O segundo objetivo trata da comparação entre os perfis axiais de temperatura do centro do cerne combustível, revestimento e fluido refrigerante da vareta mais quente do núcleo no início do ciclo de vida do reator, obtidos numericamente utilizando o acoplamento OpenMC/FiPy do módulo de ajuste termohidráulico do SPARK-NC, com aqueles encontrados em [1]. Tal comparação pode ser observada na Figura 4.8 que mostra as curvas de comparação, obtidas em vários ciclos de ajuste termohidráulico durante a primeira simulação para a obtenção de k_{eff} relativa a $t = 0$, isto é, no início do ciclo de vida do reator.

Os erros percentuais médios para os três perfis axiais ao final dessa simulação foram, conforme discutido na Seção 4.2, de 0.66% para o centro da vareta, 4.12% para o revestimento e 3.68% para o LBE circundante. Dado que o artigo original [1] não fornece nenhuma informação sobre o modelo termohidráulico utilizado, a metodologia de solução numérica, o perfil axial de distribuição de densidade volumétrica de potência da vareta e as condições de contorno impostas no domínio (ver Seção 5.3), considera-se que se alcançou um resultado razoável, visto que trata-se de um modelo assumido pelo autor ao se basear em hipóteses gerais sobre a natureza do comportamento termohidráulico de sistemas refrigerados a metais líquidos, além de se inferir o formato e magnitude do perfil axial de distribuição de densidade volumétrica de potência da vareta a partir de suposições sobre a neutronics na região dos elementos combustíveis que relacionam tal perfil com o perfil axial médio nessa região, sendo a validade aproximada dessas suposições também mostradas na Seção 4.2.

O terceiro objetivo refere-se à comparação da curva de evolução temporal do valor de k_{eff} encontrada em [1] com aquela obtida através da simulação do modelo do SPARK-NC no OpenMC, utilizando o módulo de depleção em conjunção com o módulo de ajuste termohidráulico. Os resultados podem ser vistos na Tabela 4.2, que mostra os valores de k_{eff} da simulação e do artigo em cada estágio temporal entre $t = 0$ e $t = 8.4$ anos, além da diferença desses valores em pcm, e a Figura 4.1, que confronta as curvas de evolução de k_{eff} para a simulação e para o artigo. A distribuição do erro relativo por cada estágio da depleção pode ser vista na Figura 4.2. Apesar do erro relativo se manter acima do erro devido ao desvio padrão, a média desse erro foi de 0.11%, um valor que pode ser considerado como razoável em termos de comparação de curvas. Além disto, o desvio médio foi de 121 pcm.

O quarto objetivo diz respeito à comparação entre a variação de massa dos isótopos ^{235}U , ^{238}U e ^{239}Pu devido à depleção do núcleo entre $t = 0$ e $t = 8$ anos apontadas em [1] com o modelo do

SPARK-NC no OpenMC. Na Tabela 4.2 é possível notar que a média do erro percentual nas massas foi de 0.79%, tendo o maior erro na diferença do ^{235}U , que foi de 1.74%. Tais valores também são considerados como uma aproximação bastante razoável para a diferença de massa mostrados no artigo original [1], inclusive quando se considera o isótopo ^{235}U , cuja maior diferença pode ser possivelmente explanada por esperadas diferenças entre o SARAX e o OpenMC na implementação do algoritmo e no arquivo de cadeia de depleção utilizadas, além de discrepâncias entre a densidade inicial especificada para o ^{235}U no modelo deste trabalho e o modelo do artigo original, pois este dado não é fornecido em [1] (ver Seção 5.3).

O quinto objetivo faz a comparação da curva do perfil axial de potência relativa em todo o núcleo no início do ciclo de vida do reator obtida no modelo do SPARK-NC no OpenMC com a mesma curva em [1]. Observa-se, conforme a Seção 4.3, um erro relativo médio de 3.8%, cuja maior contribuição ao dá-se no ponto mais inferior do cerne ($z = 30\text{cm}$), sendo uma possível explicação as diferenças de modelagem dos refletores/plenum entre a simulação do presente trabalho e aquela feita em [1].

Faz-se a ressalva que é impossível inferir o grau de similaridade entre a simulação feita pelos autores em [1], que originou a curva de depleção no artigo, com aquela realizada pelo autor neste trabalho; além da omissão do artigo em relação à certas composições materiais do núcleo, mencionada no primeiro parágrafo desta seção, os autores em [1] tampouco fornecem as condições de contorno da simulação neutrônica. Adicionalmente, conforme já mencionado, existe a possibilidade, bastante concebível, de que hajam diferenças entre o SARAX e o OpenMC no que se refere à implementação numérica do algoritmo de depleção e o arquivo de cadeia de depleção (*depletion chain file*) utilizados.

Além da proximidade encontrada entre as curvas de depleção, observa-se também que o cálculo de depleção no modelo do OpenMC forneceu uma diferença de massa próxima àquela do artigo original [1] (0.79% de erro médio), e o formato/magnitude do perfil axial de densidade volumétrica de potência encontrado no modelo do OpenMC, quando inserido em um código de solução numérica no domínio de uma vareta combustível, apresentou um resultado satisfatório para os três perfis axiais de temperatura mostrados no artigo [1], apesar da ausência de informações sobre o modelo térmico original (ver Seção 5.3). Desta forma, tais dados indicam que o modelo neutrônico e termohidráulico do SPARK-NC desenvolvido no OpenMC/FiPy guarda forte similaridade com o modelo desenvolvido em [1], apesar das limitações inerentes à ausência de informações sobre alguns aspectos da modelagem e simulação no artigo.

5.3 – Dificuldades e limitações

Conforme mencionado na Seção 5.2, apesar do artigo original [1] que descreve o SPARK-NC ser rico em dados, incluindo a descrição geométrica completa do núcleo, os materiais de todos os componentes (incluindo as taxas de enriquecimento dos combustíveis) e alguns parâmetros de operação do núcleo, como a potência térmica total, a velocidade média e a temperatura de entrada do fluido refrigerante, algumas informações necessárias para se reproduzir simulações de modo totalmente equivalente foram omitidas. Apesar dos materiais que integram os componentes no núcleo terem sido indicados, dados como a densidade e composição elementar dos mesmos tiveram de ser inferidas pelo autor. Como exemplo, o aço T91, que permite uma faixa relativamente ampla de composições envolvendo diferentes teores de metais, além de possíveis inclusões de impurezas, não tem seus elementos mencionados, tendo induzido o autor a buscar a composição presente em uma dada referência (ver Tabela 3.2).

Outra limitação óbvia diz respeito ao código em si: conforme mencionado na Seção 5.2, os autores em [1] não fornecem nenhum detalhe sobre o modelo termohidráulico empregado na simulação, além de nenhuma descrição das condições de contorno e hipóteses empregadas nas simulações neutrônica e termohidráulica. Além disto, os códigos que implementam tais cálculos no SARAX provavelmente apresentam discrepâncias com os códigos utilizados neste trabalho, que utilizam os módulos de depleção do OpenMC e um módulo de ajuste termohidráulico criado pelo autor baseado em um modelo de difusão-convecção em com solução por volumes finitos, além de um método de acoplamento neutrônico/termohidráulico que emprega expansões funcionais. De fato, deve ser mencionado que o SARAX é um código com abordagem determinística multigrupo, ao passo que o OpenMC é inteiramente baseado no método de Monte Carlo com dados contínuos de seções de choque, o que, por si só, pode induzir a diferenças de aproximação numérica. Por fim, uma limitação importante diz respeito ao arquivo de cadeia de depleção (*depletion chain file*, ver Seção 3.3), que não é especificada em [1] e pode levar a diferenças importantes no cálculo da evolução temporal do k_{eff} e na diferença de massa de isótopos entre estágios do processo de depleção.

5.4 – Direcionamentos Futuros

O modelo termohidráulico de difusão-convecção utilizado neste trabalho utiliza uma velocidade média do LBE como entrada, pois é um valor dado em [1]. Como um direcionamento futuro, o autor propõe a aplicação do modelo acoplado neutrônico/termohidráulico proposto neste trabalho a um outro reator, cujas curvas térmicas e de depleção são conhecidas, porém com a adição

de um código para cálculo mais preciso da parte hidráulica, envolvendo a simulação numérica de fluidos no domínio (aqui, assume-se que as condições de contorno termohidráulicas do reator são conhecidas). Tal adição pode ser feita utilizando as próprias funções do FiPy, que permitem solucionar as equações de conservação de massa e de *momentum* necessárias ao modelo. Deste modo, a validação deste modelo modificado pode ser feita através de comparação entre as curvas termohidráulicas correspondentes.

Outra proposta de direcionamento futuro está relacionada aos diversos coeficientes de reatividade do SPARK-NC, que são dadas em [1], mas não foram abordados neste trabalho. Deve ser mencionado que, para alguns coeficientes relacionados à expansão axial e radial do núcleo, os autores em [1] utilizaram uma chamada *teoria de densidade virtual*, cujos detalhes podem ser encontrados em [2]. Assim, o autor desta dissertação propõe que tal teoria seja incorporada no contexto do modelo apresentado em seu trabalho, para que tais coeficientes sejam calculados e devidamente comparados.

Por fim, propõe-se que o módulo termohidráulico proposto para o SPARK-NC possa ser aprimorado, realizando um cálculo mais preciso da distribuição de temperatura nas varetas não combustíveis (refletores, blindagem, etc.) e levando em consideração outros aspectos, como por exemplo a possível difusão térmica na direção radial das temperaturas calculadas em uma dada vareta para as varetas circunvizinhas.

5.5 – Considerações Finais

Este trabalho realizou uma modelagem neutrônica e termohidráulica de um reator rápido refrigerado à chumbo-bismuto eutético, descrito originalmente no artigo [1], fazendo comparações de certos parâmetros obtidos no modelo com aqueles mostrados no artigo original. Essa comparação apresentou resultados de erro relativo razoáveis, indicando que o modelo proposto teve êxito em incorporar, em certa medida, as características físicas do reator descrito em [1], não obstante as limitações associadas com o desconhecimento sobre alguns dados e modelos teóricos empregados para a descrição do núcleo no artigo original. É importante mencionar que o modelo proposto pelo autor utiliza exclusivamente códigos gratuitos e abertos (OpenMC para a parte neutrônica e FiPy para a parte termohidráulica), sendo ambos bibliotecas pertencentes ao ambiente *Python*. Tal empreendimento é duplamente vantajoso, pois o compartilhamento de ambiente computacional permite que o acoplamento multifísico seja feito de maneira natural, principalmente no que se refere ao intercâmbio de dados entre as partes, e a gratuidade dos códigos permitem sua ampla e pronta aplicação, sem impedimentos relativos à custos e possíveis restrições à utilização de licenças. O

autor possui expectativas que a dissertação aqui apresentada possa gerar frutos, induzindo novos trabalhos relacionados na área de engenharia de reatores no âmbito do IPEN.

5.6 – Referências do Capítulo

[1] – Hashim, M.; Cao, L.; Zhou, S.; Ma, R.; Shao, Y.; Chen, R. **SPARK-NC: A Lead-Bismuth-Cooled Small Modular Fast Reactor with Natural Circulation and Load Following Capabilities**. *Energies*, Vol. 13, 5410, 21 páginas, 2020.

[2] - Zheng, Y.; Xiao, Y.; Wu, H. **Application of the virtual density theory in fast reactor analysis based on the neutron transport calculation**. *Nuclear Engineering Design*, Vol. 20, 7 páginas, 2017.

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Diretoria de Pesquisa, Desenvolvimento e Ensino
Av. Prof. Lineu Prestes, 2242 – Cidade Universitária CEP: 05508-000
Fone/Fax(0XX11) 3133-8908
SÃO PAULO – São Paulo – Brasil
<http://www.ipen.br>

O IPEN é uma Autarquia vinculada à Secretaria de Desenvolvimento, associada à Universidade de São Paulo e gerida técnica e administrativa pela Comissão Nacional de Energia Nuclear, órgão do Ministério da Ciência, Tecnologia, Inovações e Comunicações