

Aplicação do Método de Diferenças Finitas na Equação da Difusão de Nêutrons Unidimensional

BRUNO CUNHA BARTZ¹, CLÁUDIO ZEN PETERSEN²,
MARCELO SCHRAMM³,

¹*Universidade Federal de Pelotas - brunocunhabartz@gmail.com*

²*Universidade Federal de Pelotas - claudiopetersen@yahoo.com.br*

³*Universidade Federal de Pelotas - schrammmarcelo@gmail.com*

1. INTRODUÇÃO

Foi em 1942 que o primeiro reator nuclear, o Chicago Pile 1, conseguiu realizar uma série de reações em cadeia e entrar em funcionamento. Desde o ocorrido, o interesse de pesquisadores ao redor do mundo voltou-se à energia nuclear, maravilhados com a promessa da enorme quantidade de energia contida no núcleo de um átomo (DUDERSTADT, 1976). A prova disso é que, atualmente, existem 436 usinas nucleares distribuídas entre 34 países, 16 destes dependendo da energia nuclear para suprir um quarto de suas necessidades elétricas, além de encontrarem-se em construção 53 outras usinas, dentre as quais uma é localizada no Brasil, a chamada Angra 3 (GUIMARÃES, 2010).

O princípio de funcionamento de uma usina nuclear é muito semelhante a usinas termoelétricas: uma fonte de calor aquece um líquido ou gás a altas temperaturas. O gás então se expande, girando as pás de uma turbina que converte a energia mecânica gerada em energia elétrica (REUSS, 2008). O diferencial de uma usina nuclear é que o calor utilizado para aquecer o fluido que irá mover a turbina é obtido por meio da energia liberada ao dividir-se o núcleo de um átomo que possui muitos prótons e nêutrons em dois núcleos menores através do impacto de um nêutron, um fenômeno chamado de fissão nuclear. Logo, as energias que estavam envolvidas na união deste núcleo são então liberadas na forma de calor (CARDOSO, 2012).

Cada fissão nuclear, além dos dois núcleos menores, gera também de dois a três outros nêutrons, que irão partir e fissionar outros núcleos, dando início a uma reação em cadeia (CARDOSO, 2012). O modo como uma reação em cadeia irá proceder pode ser descrito conforme um fator k , onde k é o número médio de nêutrons que são diretamente oriundos de um nêutron localizado previamente no sistema (REUSS, 2008). Para que um reator funcione a uma taxa constante, sem que as fissões nucleares cresçam a uma taxa desordenada, é necessário manter k o mais próximo possível de 1.

É por este motivo que a pesquisa na área nuclear tem se concentrado no controle da evolução populacional de nêutrons em sistemas nucleares, com destaque aos métodos analíticos, pois permitem verificar se as equações cinéticas dos nêutrons são corretamente resolvidas pelos algoritmos numéricos, permitindo ainda que se adicione os limites da equação (DULLA et al., 2007).

O presente trabalho teve como objetivo o desenvolvimento de um algoritmo com maior precisão para calcular a população de nêutrons em reatores nucleares com um controle legítimo de erro.

2. METODOLOGIA

O estudo foi realizado no município de Pelotas, localizado no Rio Grande do Sul, no período de agosto de 2018 a agosto de 2019.

Inicialmente, o código foi construído para controlar um grupo de nêutrons em um meio completamente homogêneo, através da resolução da equação abaixo (1) contendo Problemas de Valor de Contorno (PVC) em $\emptyset(0)=0$ e $\emptyset(L)=0$, onde L representa o x máximo da malha utilizada:

$$-\emptyset''(x) + \emptyset(x) = \frac{1}{k} \cdot 0,5 \cdot \emptyset(x) \quad (1)$$

Em um segundo caso, desta vez mais completo, a equação cinética dos nêutrons utilizada para construir o código foi a equação (2), com PVC idênticos ao caso anterior:

$$-D(x)\emptyset''(x) + \Sigma_A(x)\emptyset(x) = \frac{1}{k} \cdot \nu\Sigma_F \cdot \emptyset(x) \quad (2)$$

onde:

\emptyset : fluxo escalar de nêutrons;

D : coeficiente de difusão de nêutrons;

Σ_A : seção de choque macroscópica de absorção;

k : criticalidade;

ν : número médio de nêutrons emitido por fissão;

Σ_F : seção de choque macroscópica de fissão.

Para este caso, o código foi desenvolvido de modo a simular o ambiente heterogêneo no interior de um sistema nuclear para um único grupo de nêutrons, com um total de três regiões distintas, nas quais D , Σ_A e $\nu\Sigma_F$ possuem diferentes valores. Foi então empregado o princípio analítico de *benchmarking*.

Para a resolução do PVC foi utilizado o Método das Diferenças Finitas (MDF), que consiste no rearranjo de um problema contínuo em um problema discreto ao utilizar fórmulas de diferenças finitas na malha desejada. A resolução do problema discreto é feita escrevendo-a na forma matricial $A\emptyset = b$, onde A é uma matriz tridiagonal de coeficientes. No caso (2), para a resolução desta matriz A , foi escolhido o algoritmo de Thomas (TDMA) (MALISKA, 2017), por este método provar-se mais adequado e requerir menos tempo de processo. Através de deduções, foram obtidas as seguintes definições dos coeficientes da matriz A e também da matriz b :

$$\begin{aligned} a_{wj}^{(r)} &= -2D^{(r-1)} \\ a_{pj}^{(r)} &= 2D^{(r-1)} + 2D^{(r)} + dx^2(\Sigma_A^{(r-1)} + \Sigma_A^{(r)}) \\ a_{ej}^{(r)} &= -2D^{(r)} \\ b_j &= \frac{1}{k} \cdot dx^2 \cdot (\nu\Sigma_F^{(r-1)} + \nu\Sigma_F^{(r)}) \cdot \emptyset^{(r)} \end{aligned}$$

Onde (r) indica em qual região da malha os parâmetros localizam-se.

Para respeitar o estabelecimento das condições de contorno, o primeiro e último termos da matriz A foram forçadas a zero, assim como a primeira e última linhas de b . Seguindo, a cada iteração realizada o código atualiza os valores do vetor \emptyset e, portanto,

também de sua criticalidade (k), que encontra-se dentro de b . A forma como k é atualizado pode ser vista na Equação 3, abaixo.

$$k_{i+1} = k_i \cdot \frac{\int_0^L \nu \Sigma_F(x) \cdot \emptyset_{i+1}(x) \cdot dx}{\int_0^L \nu \Sigma_F(x) \cdot \emptyset_i(x) \cdot dx} \quad (3)$$

Onde o subíndice i representa a i -ésima iteração.

Ainda no caso heterogêneo, ao executar o código o usuário é questionado acerca de três preferências: 1) quão refinada ele deseja que a malha seja; 2) qual deve ser o critério de parada, baseado no valor do erro; e 3) qual o caso desejado.

O código se encerra apenas quando o erro (Equação 4) é menor que o critério definido pelo usuário na preferência 2.

$$\text{erro} = (k_i - k_{i-1})/k_i \quad (4)$$

3. RESULTADOS E DISCUSSÃO

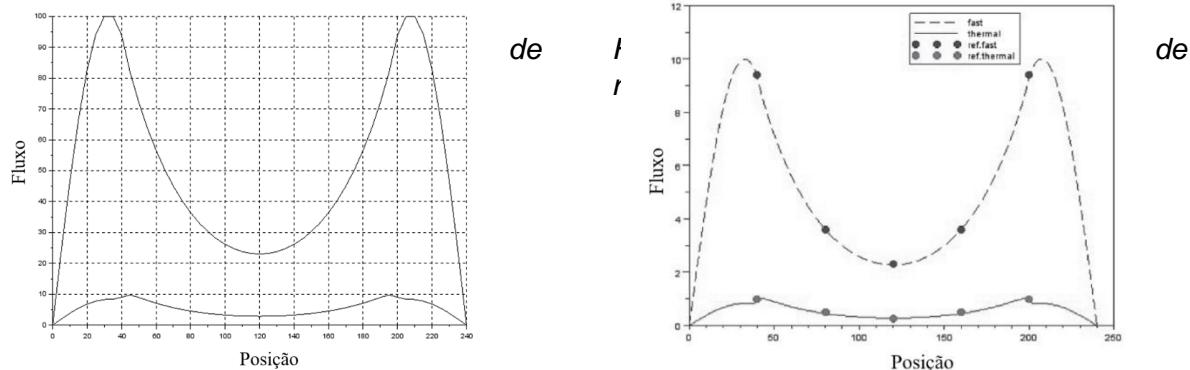
Para mostrar os resultados em comparação com trabalhos já realizados anteriormente, optou-se por adotar os mesmos parâmetros utilizados por CEOLIN (2014) (Tabela 1). O código foi executado em um computador com sistema operacional Windows 10 (64-Bit), processador Intel Core i3-6100U de 2,30 GHz, 4Gb de memória RAM e utilizando-se o programa SciLab versão 5.5.2.

	$D_g^{[i]}(\text{cm}^{-1})$	$\Sigma_{rg}^{[i]}(\text{cm}^{-1})$	$\nu \Sigma_{fg}^{[i]}(\text{cm}^{-1})$	$\Sigma_{s12}^{[i]}(\text{cm}^{-1})$
Região	$g = 1$	$g = 2$	$g = 1$	$g = 2$
1 e 3	1,5	0,5	0,026	0,18
2	1,0	0,5	0,02	0,08

Tabela 1. Parâmetros utilizados pelo Argonne National Laboratory

A localização das interfaces é determinada pelo método de *benchmarking*, onde o código é capaz de abrigar vários casos com diferentes parâmetros e pontos de interface. Assim, o usuário pode adotar o caso que lhe seja mais conveniente. Neste caso, o *benchmarking* utilizado por CEOLIN (2014) foi disponibilizado pelo Argonne National Laboratory, um centro de pesquisa em engenharia e ciência multidisciplinar localizado nos Estados Unidos da América. Neste caso temos um $L = 240 \text{ cm}$, com fronteiras em $L = 40 \text{ cm}$ e em $L = 200 \text{ cm}$.

Com estes parâmetros inseridos no código, este gerou o gráfico mostrado na Figura 1. Ao lado deste, pode-se ver o gráfico gerado pelo código criado por [2] (Figura 2). O critério de parada adotado foi o mesmo de [2], $\text{erro} < 10^{-15}$.



Segundo o código de [2], com $dx = 10 \text{ cm}$, ao final k convergiu para 0,90155. Na Tabela 2 abaixo mostramos os resultados que obtivemos com o código construído aplicando três diferentes refinamentos de malha.

Refinamento	$dx (\text{cm})$	Tempo de processo (s)	Número de iterações	k
1	10	86,08	14662	0,901119
5	2	1300,89	128008	0,901551
10	1	8583,80	406187	0,901584

Tabela 2. Resultados obtidos utilizando três diferentes refinamentos de malha

4. CONCLUSÕES

Com relação à comparação realizada com CEOLIN (2014), o resultado foi satisfatório, uma vez que o valor para o qual k convergiu em ambos os códigos mostrou-se semelhante. Todavia, é importante destacar a sensibilidade do código ao aumento do refinamento da malha, gerando dx cada vez menores.

Os resultados observados ao variarmos dx constituem um ponto fraco do código, que acaba tornando o processo mais demorado a medida que exigimos um refinamento maior dele.

Os próximos objetivos da pesquisa seriam deixar os códigos mais próximos de situações reais com multigrupos de nêutrons e em regimes que sejam transientes, não apenas uni, mas bidimensionais e tridimensionais.

5. REFERÊNCIAS BILIOGRÁFICAS

CARDOSO, E. M. **A Energia Nuclear**. 3.ed. Rio de Janeiro: CNEN, 2012. (Apostila Educativa).

CEOLIN, C. **Solução Analítica da Equação Cinética de Difusão Multigrupo de Nêutrons em Geometria Cartesiana Unidimensional pela Técnica da Transformada Integral**. 2010. Dissertação de Mestrado – UFRGS, Porto Alegre/RS.

DUDERSTADT, J; HAMILTON, L. **Nuclear Reactor Analysis**. New York: John Wiley & Sons, 1976.

DULLA, S.; RAVETTO, P.; PICCA, D; TOMATIS, D. **Analytical Benchmarks for the Kinetics of Accelerator-driven Systems**. Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications. Monterey/CA, 2007.

REUSS, Paul. **Neutron Physics**. France: EDP Sciences, 2008.

GUIMARÃES, L. **A indústria nuclear hoje**, Valor Econômico, São Paulo, 2010.

MALISKA, C. R. **Transferência de Calor e Mecânica dos Fluídos Computacional**, 2.ed., LTC, Rio de Janeiro, 2017.