Efficiency and implementation of alternating direction implicate method for neutron diffusion equation in three-dimensional space

P A Pugachev, A M Sirotkin, V I Romanenko and R B Bahdanovich
National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Kashirskoe shosse 31, Moscow, Russia
E-mail: pahapugach@mail.ru

Abstract. The paper presents algorithm of solving neutron diffusion equation in three-dimensional space based on multiple application of alternating direction implicit method. The point of the method is to reduce solving of multidiagonal systems of equations to solving of few three diagonal systems in iterative processes. Diffusion code CriMR that uses this algorithm implemented for triangular mesh is described. Results of initial tests of the code are also presented (comparison of parameters of three different homogenized cores of BN-600 calculated by CriMR with ones of MCU- PTR).

1. Introduction
With each year efficiency of modern computer facilities increases, so this fact allows us in many cases to perform calculations without thinking of performance quality. Therefore, optimization and efficiency of utilized codes should not be underestimated. Despite expansion of Monte-Carlo codes, diffusion codes are not forgotten and still developing [1-2].

The aim of this work was to create high-performance diffusion code based on multiple application of alternating direction implicit method for solving neutron diffusion equation in triangular three-dimensional mesh.

2. Method
The alternating direction implicit method (ADI) [3] is one of the methods of solving systems of linear equations. ADI allows reducing of solving system (1) to solving of a number of less diagonal systems of equations.

\[ \hat{A}\vec{\varphi} = \vec{q} \] (1)

To implement ADI, matrices B and C (where \( \hat{B} + \hat{C} = \hat{A} \)) must be determined. If matrices \( \hat{A} \), \( \hat{B} \) and \( \hat{C} \) are symmetric positive-definite matrices, we can implement ADI to solve system (1) via iterative process (2).

\[ (\hat{A} + \omega_1 \hat{E})\vec{\varphi}^{t+\frac{1}{2}} = \vec{q} - (\hat{B} - \omega_1 \hat{E})\vec{\varphi}^{t} \] (2)
\[ (\hat{B} + \omega_2 \hat{E})\vec{\varphi}^{t+1} = \vec{q} - (\hat{A} - \omega_2 \hat{E})\vec{\varphi}^{t+\frac{1}{2}} \]
It is supposed that systems of process (2) can be resolved much easily then initial one. Wherein rate of convergence depends on \( \omega_1 \) and \( \omega_2 \) coefficients determined by spectral bounds of corresponding matrices (3).

\[
\omega_1 = \left[ (\lambda_{\text{min}}(\hat{A}) - \theta)(\lambda_{\text{max}}(\hat{A}) - \theta) \right]^{1/2} - \theta
\]
\[
\omega_2 = \left[ (\lambda_{\text{min}}(\hat{B}) + \theta)(\lambda_{\text{max}}(\hat{B}) + \theta) \right]^{1/2} + \theta
\]

where \( \theta = \frac{\lambda_{\text{min}}(\hat{A})\lambda_{\text{max}}(\hat{B}) - \lambda_{\text{min}}(\hat{B})\lambda_{\text{max}}(\hat{A})}{\lambda_{\text{min}}(\hat{A})+\lambda_{\text{max}}(\hat{A})+\lambda_{\text{min}}(\hat{B})+\lambda_{\text{max}}(\hat{B})} \)

2.1. Finite-difference transformation of diffusion equation

Let’s consider equation (4) – non-homogeneous diffusion equation with set of boundary conditions and domain with triangular mesh (figure 1).

\[-\text{div}(D\text{grad}\varphi) + \Sigma \varphi = Q, \quad \frac{\partial \varphi}{\partial n} + \gamma_2 D \varphi = 0 \] (4)

where \( \gamma \) shows which type of boundary condition is set: \( \gamma = 0 \) : reflection, \( \gamma = \infty \) : \( \varphi(\text{real boundary}) = 0 \), \( \gamma = \infty \) : \( \varphi(\text{extrapolated boundary}) = 0 \).

![Figure 1. Horizontal layer of three-dimensional mesh.](image)

Carry on finite-difference transformation of equation (4) by integrating it in primitive cell (triangular prism). First member of equation (4) after integrating can be transformed according to Green’s theorem to \( \int_S \frac{\partial \varphi}{\partial n} dS \) which we can consider as sum of five surface integrals. For example, one of these integrals (corresponded to upper bound of prism) is taken as follows:

\[
\int_{S_v} D \frac{\partial \varphi}{\partial n} dS = D_{i,j,k} S_v \frac{\varphi_{i,j,k+1} - \varphi_{i,j,k}}{h_v}
\]
Remaining four integrals can be dealt with the same way. Other members of equation (4) are much easily to integrate:

\[
\int_\Omega \Sigma \varphi \, dv = \Sigma_{i,j,k} \varphi_{i,j,k} V, \quad \int_\Omega Q \, dv = Q_{i,j,k} V
\]

After finite-difference transformation, we will get the seven diagonal system of equations (5) to be resolved by ADI (every considering matrices are symmetric positive-definite matrices, so we can do it).

\[
a_{i,j,k} \varphi_{i-1,j,k} + b_{i,j,k} \varphi_{i,j,k} + c_{i,j,k} \varphi_{i+1,j,k} + d_{i,j,k} \varphi_{i,j-1,k} + e_{i,j,k} \varphi_{i,j+1,k} + w_{i,j,k} \varphi_{i,j,k-1} + v_{i,j,k} \varphi_{i,j,k+1} = q_{i,j,k}
\]  

(5)

2.2. Implementation of ADI

After first applying of ADI to system (5) iterative process (6) will take place.

\[
\begin{align*}
(L_{xy} + \omega_1 \hat{E}) \varphi^{t+\frac{1}{2}} &= \hat{q} - (L_x - \omega_1 \hat{E}) \varphi^{t} = \tilde{q}_z \\
(L_z + \omega_2 \hat{E}) \varphi^{t+1} &= \hat{q} - (L_{xy} + \omega_2 \hat{E}) \varphi^{t+\frac{1}{2}}
\end{align*}
\]  

(6)

In process (6) the second matrix \( L_z \) is three diagonal matrix whereas first one \( L_{xy} \) is five diagonal matrix, so first system of process (6) can be resolved via ADI too with iterative process (7) [4-5].

\[
\begin{align*}
(L_x + \left( \omega_1' + \frac{\alpha_1}{2} \right) \hat{E}) \varphi^{s+\frac{1}{2}} &= q_z - (L_y - \left( \omega_1' + \frac{\alpha_1}{2} \right) \hat{E}) \varphi^s \\
(L_y + \left( \omega_2' + \frac{\alpha_2}{2} \right) \hat{E}) \varphi^{s+1} &= q_z - (L_x - \left( \omega_2' + \frac{\alpha_2}{2} \right) \hat{E}) \varphi^{s+\frac{1}{2}}
\end{align*}
\]  

(7)

Finally, every considering system is three diagonal system which we will solve by tridiagonal matrix algorithm.

In order to obtain \( \omega \) coefficients of (6)-(7) processes spectral bounds of corresponding matrices are required. The power method and bisection method are used to find spectral bounds of five and three diagonal matrices respectively.

3. Diffusion code

3.1. Algorithm

We deal with homogeneous multigroup equation. Therefore, we perform source iterations. On each iteration, set of non-homogeneous equations (from high-energy group to low) is solved by processes (6) - (7).

3.2. CriMR code

Diffusion code CriMR is based on aforementioned algorithm. Figure 2 shows references between parts of the code. It consists of the following modules:

- **CriMR** is operating shell written on Python. It allows to set input data and parameters, to process output data and also controls execution of other modules;
- **KOEFF** calculates coefficients of finite-difference transformation;
- **EIGVAL** found spectral bounds of matrices and determines \( \omega \);
- **FLUX** performs source iterations;
- **READER** prepares macro cross-sections;
- **HEAT** normalizes flux according to thermal power of reactor;
- **BURN** is module for burnup calculations;
- **SCRE** calculates reaction rates;

Except **CriMR**, modules are written on FORTRAN. Optimization and parallelization (by means of OpenMP library) were performed.

![Figure 2. Structural scheme of CriMR code.](image)

### 4. Results of testing
For initial testing of the code, simulation of three different models of BN-600 reactor [6] was performed. These models are homogenized cores with blanket zones (see figure 3). Each model consists of three core zones: F1 – zone with low enrichment, F2 – zone with average enrichment, F3 – zone with high enrichment (in hybrid core zone F4 contains UO$_2$ with high enrichment, zone F3 – MOX fuel with high enrichment, other zones is the same as for UO$_2$ fuel core). Besides there are upper and lower blanket zones for each core zone (LB and UB respectively) and radial blanket zone (RB).
The difference between models is in fuel composition: first model is with UO$_2$ fuel only, second is with MOX fuel and third one with UO$_2$ fuel and small zone with MOX fuel. Properties of models are presented in Table 1.

### Table 1. Properties of models.

| Zone | Temperature, K | Enrichment (UO$_2$ fuel only) | Pu fraction (MOX fuel only) | Pu fraction for hybrid core |
|------|----------------|-------------------------------|-----------------------------|-----------------------------|
| LB   | 650            | 0.3 %                         | --                          | --                          |
| F1   | 1500           | 17 %                          | 15.3 %                      | --                          |
| F2   | 1500           | 21 %                          | 16.9 %                      | --                          |
| F3   | 1500           | 26 %                          | 18.5 %                      | 18.5 %                      |
| F4   | 1500           | --                            | --                          | --                          |
| UB   | 823            | 0.3 %                         | --                          | --                          |
| RB   | 736            | 0.3 %                         | --                          | --                          |

CriMR calculations were performed in 26 energy groups with 0.1E-6 accuracy for source iterations, 0.1E-7 and 0.1E-8 for outer and inner ADI iterations respectively.

### 4.1. Choosing of optimal partition

In order to choose optimal number of points of spatial discretization for CriMR calculation test series for each model were performed. Optimum means such number of points at which in case of further increase of the number observed parameter ($K_{eff}$) changes slightly. As shown in Figure 4, results for different models are almost the same: 0.18E+6 points is optimum case. It is also notable that dependence of running time on number of points is linear, which obvious due to possibility of elementwise (non-matrix) operations in processes (6) - (7).
Figure 4. $K_{eff}$ /running time – number of points dependences, (a) – MOX core, (b) – UO2 core, (c) – Hybrid core.
4.2. Results
After choosing optimal number of points, series of calculation with CriMR and MCU-PTR [7] were performed to obtain and compare interesting for us parameters. These parameters are:

- $K_{\text{eff}}$;
- sodium void effect for whole reactor;
- $\text{RF5/RF8}$ (fission rate of $^{235}U$ to fission rate of $^{238}U$) ratio for every physical zone of core including blanket zones.

Results for $K_{\text{eff}}$ are shown in table 2. Despite of different nuclear data libraries, results are close enough.

| Table 2. $K_{\text{eff}}$ for three models. |
|------------------------------------------|
| Hydor core | MOX core | UO$_2$ core |
|------------|----------|-------------|
| MCU        | CriMR    | MCU        | CriMR    | MCU        | CriMR    |
| $K_{\text{eff}}$ | 1.00059  | 0.99733  | 0.99912  | 0.99683  | 0.99835  | 0.99480  |
| std. dev.  | 0.00004  | --       | 0.00006  | --       | 0.00005  | --       |
| del        | --       | -0.00326 | --       | -0.00229 | --       | -0.00354 |

As well as the results for $\text{RF5/RF8}$ ratio (see table 3); residual ((MCU-CriMR)/MCU, %) for whole reactor does not exceed 2.5% that can be considered satisfactory.

| Table 3. RF5/RF8 ratio for three models |
|----------------------------------------|
| Hydor core | MOX core | UO$_2$ core |
|------------|----------|-------------|
| MCU        | CriMR    | MCU        | CriMR    | MCU        | CriMR    |
| RF5/RF8    | 4.27E-01 | 4.51E-01  | 5.49      | 4.24E-01 | 4.56E-01  | 7.63      | 4.26E-01 | 4.48E-01 | 5.02 |
| RF5/RF8    | 6.92E-01 | 4.03E-01  | 4.37      | 4.05E-01 | 4.30E-01  | 6.02      | 3.79E-01 | 3.93E-01 | 3.56 |
| RF5/RF8    | 3.87E-01 | 3.98E-01  | 2.87      | 4.13E-01 | 4.29E-01  | 3.90      | 3.68E-01 | 3.75E-01 | 1.71 |
| RF5/RF8    | 8.42E+00 | 8.56E+00  | 1.61      | 1.25E-01 | 1.31E-01  | 4.58      | 8.40E+00 | 8.52E+00 | 1.42 |
| RF5/RF8    | 5.04E+00 | 9.91E+00  | 1.92      | 1.20E+01 | 1.24E+01  | 3.62      | 9.53E+00 | 9.64E+00 | 1.06 |
| RF5/RF8    | 1.04E+01 | 1.07E+01  | 3.01      | 1.16E+01 | 1.22E+01  | 5.19      | 1.13E+01 | 1.17E+01 | 3.00 |
| RF5/RF8    | 7.38E+00 | 1.31E+01  | 5.28      | --       | --        | --        | --       | --       | --   |
| RF5/RF8    | 3.27E+01 | 3.26E-01  | 0.29      | 3.23E-01 | 3.30E-01  | 2.02      | 3.27E-01 | 3.25E-01 | 0.43 |
| RF5/RF8    | 2.82E+01 | 2.83E-01  | 0.59      | 2.91E-01 | 2.98E-01  | 2.27      | 2.77E-01 | 2.77E-01 | 0.00 |
| RF5/RF8    | 3.26E+01 | 3.36E-01  | 3.07      | 3.46E-01 | 3.60E-01  | 4.16      | 3.10E-01 | 3.16E-01 | 1.91 |
| Reactor    | 6.35E+00 | 6.49E+00  | 2.16      | 1.71E+01 | 1.73E+01  | 1.44      | 7.71E+00 | 7.81E+00 | 1.37 |

Results for SVE (table 4) don’t agree good enough but MCU’s part of them were obtained with quite low statistics comparing to statistics used for initial state.
Table 4. Sodium void effect for three models.

|                  | Hybryd core | MOX core | UO₂ core |
|------------------|-------------|----------|----------|
|                  | MCU | CriMR | del, % | MCU | CriMR | del, % | MCU | CriMR | del, % |
| reactor          | -0.00794 | -0.00774 | 2.48 | 0.01544 | 0.01616 | 4.62 | -0.00734 | -0.00931 | 26.73 |

5. Conclusion
During this work, algorithm based on multiple implementation of alternating direction implicit method was developed. Diffusion code CriMR using this algorithm was created. Initial testing of CriMR code was performed via simulation of three different models of BN-600 cores, then the results were compared with obtained by MCU-PTR. Comparison shown good agreement between CriMR and MCU results in calculation of $K_{ef}$ and fission reaction rates ratio, especially considering different nuclear data libraries and diffusion theory assumptions, but results for SVE does not show good agreement and obvious reason for it is low statistic used in MCU-PTR for simulation of these cases.

Acknowledgements
This work was supported by Competitiveness Program of National Research Nuclear University MEPhI.

References
[1] Galicia-Aragón J et al. 2018 Initial verification of AZNHEX hexagonal-$z$ neutron diffusion code with MCNP6 for two different study cases *Progress in Nuclear Energy* **106** pp 284-292
[2] Fiorina C et al. 2016 Development and verification of the neutron diffusion solver for the GeN-Foam multi-physics platform *Annals of Nuclear Energy* **96** pp 212-222
[3] Douglas J 1962 Alternating direction methods for three space variables *Numerische Mathematik* **4** pp 41-63
[4] Wachspress E L 1994 Three-variable alternating-direction-implicit iteration *Computers math. appl.* **27**(3) pp. 1-7
[5] Wachspress E L 1984 Generalized ADI preconditioning *Computers math. appl.* **10**(6) pp 457-461
[6] Bahdanovich R B, Romanenko V I and Tikhomirov G V 2017 Contribution of neutron-capture reactions in energy release in the fuel core of BN-600 *IOP Conf. Series: Journal of Physics: Conf. Series* **781**
[7] Alekseev N I et.al. 2011 MCU-PTR program for high-precision calculations of pool and tank type research reactors *Atomic energy* **109**(3) pp 149–156