energies

MDPI

*Article*

# Verification of the Parallel Transport Codes Parafish and AZTRAN with the TAKEDA Benchmarks

**Julian Duran-Gonzalez [1,*], Victor Hugo Sanchez-Espinoza [1], Luigi Mercatali [1], Armando Gomez-Torres [2] and Edmundo del Valle-Gallegos [3]**

[1] Karlsruhe Institute of Technology (KIT), Institute of Neutron Physics and Reactor Technology (INR), Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany; victor.sanchez@kit.edu (V.H.S.-E.); luigi.mercatali@kit.edu (L.M.)

[2] Instituto Nacional de Investigaciones Nucleares, Departamento de Sistemas Nucleares, Carretera México-Toluca S/N, Ocoyoacac 52750, Mexico; armando.gomez@inin.gob.mx

[3] Instituto Politécnico Nacional, Escuela Superior de Física y Matemáticas, Av. IPN S/N, Alcaldía Gustavo A, Madero 07738, Mexico; edmundo.delvalle@gmail.com

* Correspondence: julian.gonzalez@kit.edu

**Abstract:** With the increase in computational resources, parallel computation in neutron transport codes is inherent since it allows simulations with high spatial-angular resolution. Among the different methodologies available for the solution of the neutron transport equation, spherical harmonics ($P_N$) and discrete-ordinates ($S_N$) approximations have been widely used, as they are established classical methods for performing nuclear reactor calculations. This work focuses on describing and verifying two parallel deterministic neutron transport codes under development. The first one is the Parafish code that is based on the finite-element method and $P_N$ approximation. The second one is the AZTRAN code, based on the RTN-0 nodal method and $S_N$ approximation. The capabilities of these two codes have been tested on the TAKEDA benchmarks and the results obtained show good behavior and accuracy compared to the Monte Carlo reference solutions. Additionally, the speedup obtained by each code in the parallel execution is acceptable. In general, the results encourage further improvement in the codes to be comparable to other well-validated deterministic transport codes.

**Keywords:** Neutron transport equation; Spherical HARMONICS ($P_N$); Finite Element Method; Discrete Ordinates ($S_N$); RTN-0 Nodal method

check for updates

**Citation:** Duran-Gonzalez, J.; Sanchez-Espinoza, V.H.; Mercatali, L.; Gomez-Torres, A.; Valle-Gallegos, E.d. Verification of the Parallel Transport Codes Parafish and AZTRAN with the TAKEDA Benchmarks. *Energies* **2022**, *15*, 2476. https://doi.org/10.3390/en15072476

Academic Editor: Dan Gabriel Cacuci

Received: 24 February 2022
Accepted: 24 March 2022
Published: 28 March 2022

## 1. Introduction

Given the increase in computational resources, parallel computation has become a powerful tool for numerical calculations since it takes an essential role in developing neutron transport codes for nuclear reactor analysis, allowing for high-fidelity pin-by-pin full core transport analysis in a reasonable computational time, instead of the commonly used neutron diffusion calculation. Currently, several methodologies provide an excellent approximation to the neutron transport equation, each one with associated strengths and weaknesses. Of all these, some of the most widely accepted and well-established methodologies are the spherical harmonics ($P_N$) [1] and the discrete ordinates ($S_N$) [2] methods.

The $P_N$ method expands the angular dependence of the neutron flux in terms of spherical harmonics, which leads to an infinite set of coupled differential equations. This set is reduced to a finite system by truncating the expansion at an arbitrarily high order of accuracy $N$. The exact solution to the transport equation is reached for an order of $N \to \infty$. An advantage of this method is that it does not exhibit the so-called "ray effect" [3] (unphysical oscillations in the scalar flux related to other formulations, like the case of $S_N$). However, the main drawback of the $P_N$ method is that, for a 3D geometry, the number of equations increases and becomes complicated to solve, with a consequent huge demand of RAM memory. On the other hand, the $S_N$ method selects a set of angular directions with their respective weights to approximate the angular neutron flux by replacing it with a weighted sum. The $S_N$ approximation is characterized by its simplicity and versatility but suffers from the "ray effect", which can be smoothed by increasing the number of discrete-ordinates. Although, sometimes, the "ray effect" tends to be persistent, some techniques have been proposed to mitigate its impact [4].

In general, to approximate the steady-state neutron transport equation for problems with multiplying media (1) in a deterministic way, it is required to discretize its independent variables and use numerical and iterative methods to reach a solution.

$$\begin{aligned}
&\hat{\Omega}\cdot\vec{\nabla}\psi(\vec{r},E,\hat{\Omega})+\Sigma_t(\vec{r},E)\psi(\vec{r},E,\hat{\Omega})= \\
&\int_0^\infty dE' \int_{4\pi} \Sigma_s(\vec{r},E'\to E,\hat{\Omega}'\to\hat{\Omega})\psi(\vec{r},E',\hat{\Omega}')d\hat{\Omega}'+ \\
&\frac{\chi(\vec{r},E)}{k_{eff}}\int_0^\infty dE' \int_{4\pi} v\Sigma_f(\vec{r},E')\psi(\vec{r},E',\hat{\Omega}')d\hat{\Omega}',
\end{aligned} \tag{1}$$

where $\psi(\vec{r},E,\hat{\Omega})$ is the angular neutron flux, $\chi(\vec{r},E)$ is the fission neutron spectrum, $k_{eff}$ is the effective multiplication factor, $v$ is the average number of neutrons emitted per fission and $\Sigma_t, \Sigma_s$, and $\Sigma_f$ are the corresponding total, scattering, and fission macroscopic cross-sections. Additionally, the independent variables are energy $(E)$, angle $(\hat{\Omega})$, and space $(\vec{r})$. Thus, to discretize Equation (1), the following techniques are commonly used: multi-group approximation (energy), $P_N/S_N$ methods (angle), and finite element method/nodal method/diamond-difference method (space). Several deterministic codes in the literature approximate Equation (1). Some of the well-known codes based on the methods mentioned above are VARIANT [5], EVENT [6], PARTISN [7], and TORT-TD [8].

This work is focused on the verification of the predictive capabilities of two parallel deterministic codes currently under development (one based on $P_N$ and the other one on $S_N$) by means of the TAKEDA Benchmark. Numerical results from these codes are compared against the results obtained with a Monte Carlo reference code. First, the descriptions of the parallel $P_N$ and $S_N$ neutron transport codes that are under development are given in Section 2, followed by the description of the TAKEDA Benchmarks in Section 3. Then, in Section 4, their respective results are analyzed; subsequently, the parallel numerical results obtained are given in Section 5. Finally, conclusions and future work are summarized in Section 6.

## 2. Parallel Neutron Transport Codes under Development

### 2.1. Parafish

The Parafish code [9] (Parallel finite element spherical harmonic) is a parallel neutron transport solver written in C++ and developed at the Karlsruhe Institute for Technology (KIT). Parafish solves the 2D/3D (cartesian) steady-state neutron transport equation for multiplying media via the following techniques: multigroup approximation for energy discretization, the $P_N$ method to treat the angular direction applying the so-called even parity formulation, from which the even component of the angular dependence is calculated, and, finally, for the spatial discretization that uses non-conforming finite elements (FEs) [10].

The iterative process used in Parafish can be divided into the outer (to obtain the $k_{eff}$) and inner (to approximate the angular neutron flux) iterations. For the outer iterations, the Davidson method [11] is employed. This method is an alternative algorithm to the well-known Power method [12] and computes a few of the extremal eigenvalues and the corresponding eigenvectors of a large sparse real matrix. This method is based on a projection on a dedicated subspace, built iteratively. Regarding inner iterations, the coupled energy–space–angle system is solved in two parts. First, using the Gauss–Seidel method for the energy level. Then, the GMRES method [13] with block-diagonal preconditioning (incomplete Cholesky method) for the spatial-angular level.

Concerning the Parafish parallelization, a non-overlapping spatial domain decomposition scheme is applied, in agreement with modified Schwarz methods [14] with the duplication of the interface finite element nodes. The message passing interface (MPI) was selected to perform this implementation. A more detailed description is found in the following reference [9].

In the Parafish calculation scheme [11]: first, an energy–space–angle matrix is built as a tensor product of the variables. Then, this linear system is solved using the Gauss–Seidel method. At each Gauss–Seidel iteration (energy level), the diagonal block of the matrix within a given energy group corresponds to a space–angle matrix obtained by the spatial domain decomposition in which each block is symmetric. Consequently, the blocks can be treated individually by different processors. Hence, the space–angular matrices can be solved using the GMRES method with block-diagonal preconditioning (IC). Since the domains are spread over the available processors, these "local solves" are performed in parallel. Figure 1 illustrates the calculation scheme described.

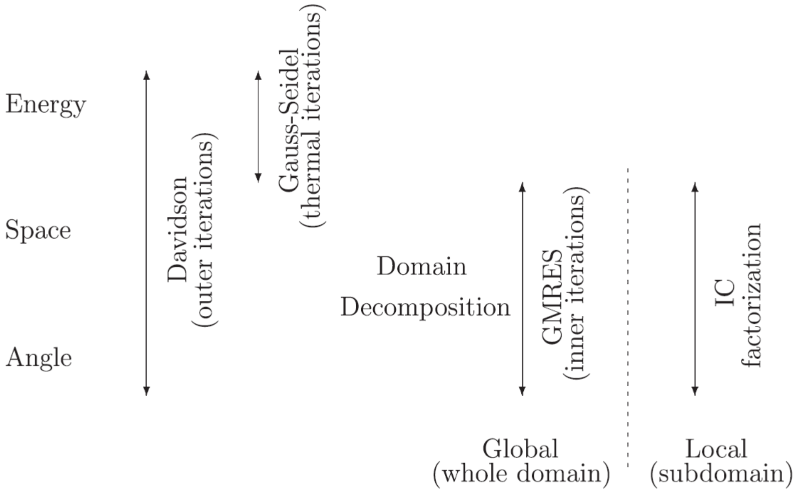

**Figure 1.** Parafish calculation scheme.

### *2.2. AZTRAN*

One of the main objectives of the Mexican AZTLAN platform project [15] is the development of in-house software for the analysis of nuclear reactors. One of the tools under development is the AZTRAN code [16] (AZtlan neutron Transport for Reactor ANalysis), which solves the steady-state neutron transport equation in 3D cartesian geometry, specifically for problems in multiplicative media.

AZTRAN is written in Fortran 90, and to approximate the neutron transport equation, the independent variables are discretized: by using multigroup approximation (energy), the $S_N$ method (angle), and a polynomial nodal method RTN-0 [17] (space). Applying these approximations in Equation (1), a linear system of equations is obtained, and iterative methods are required for its solution. In this case, the angular neutron flux approximation (inner iterations) is obtained using the source iteration method [2] (serial solution). However, when solving a steady-state problem with multiplicative media, the equation becomes an eigenvalue problem (outer iterations), so the classical Power method is applied.

For the parallel implementation, a spatial domain decomposition [18] with non-overlapping domains is employed in AZTRAN [16]. The message passing interface (MPI) library is used and provides the intrinsic functions for creating a virtual cartesian topology. In the beginning, the *MPI_CART_CREATE* function generates a 3D cartesian topology, creating a new communicator in which topology information has been attached. Next, the *MPI_CART_COORDS* function is used to calculate the corresponding coordinates associated with each MPI process. After that, the *MPI_CART_SHIFT* function is applied to get the id of the neighborhood for a specific processor. Finally, the spatial subdomains are built and then assigned to the processors from these functions. Thus, each processor is considered to solve an "independent problem"; therefore, they are solved simultaneously. Figure 2 shows the 3D-Domain decomposition employed by AZTRAN.

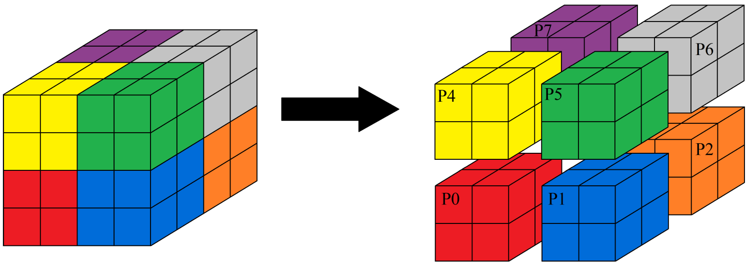

**Figure 2.** Spatial domain decomposition spreads into eight processors.

In the case of AZTRAN, the following solution scheme is followed [16]: Since the domain is splitting into non-overlapping multiple subdomains allocated to the processors, it allows the simultaneous solution of the angular neutron fluxes in each subdomain. Nevertheless, the subdomains are associated through the angular neutron flux from adjacent subdomain interfaces. Therefore, at the end of the parallel solutions, the angular neutron fluxes in the subdomain boundaries are exchanged. Consequently, the *MPI_SENDRECV* feature has been used to send and receive successive blocking communication data. This iterative process occurs until the solution converges in agreement with the specific convergence criteria selected by the user. A simplified flowchart of this procedure is shown in Figure 3.

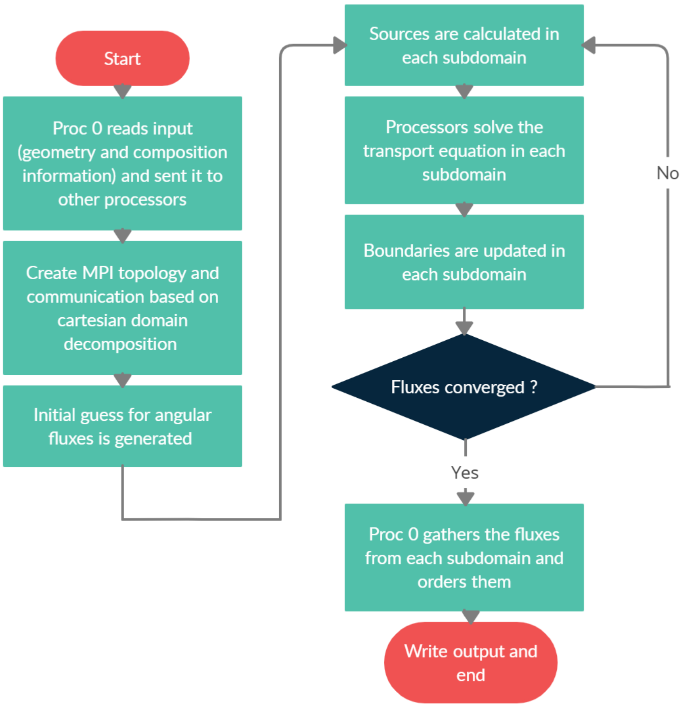
Start
Proc 0 reads input (geometry and composition information) and sent it to other processors
Create MPI topology and communication based on cartesian domain decomposition
Initial guess for angular fluxes is generated
Sources are calculated in each subdomain
Processors solve the transport equation in each subdomain
Boundaries are updated in each subdomain
Fluxes converged ?
No
Yes
Proc 0 gathers the fluxes from each subdomain and orders them
Write output and end

**Figure 3.** Flowchart of parallel AZTRAN scheme.

The iterative procedure here is not the classic Source Iteration method (iterating value on the scattering source). In this case, it also iterates on the interface fluxes between each subdomain [18], which implies increasing inner iterations to converge, leading to asynchronous parallelism. The reason is that updating the subdomain boundaries produces an iteration penalty, since it requires extra iterations to propagate the angular neutron fluxes information through the whole domain, unlike the sequential method that only requires one iteration. Although increasing subdomains increases inner iterations, and this can degrade parallel performance, good acceleration can be achieved and was demonstrated in [16].

## 3. TAKEDA Benchmarks Description

Three 3D TAKEDA benchmark problems were selected for the present study, compiled by Takeda and Ikeda in the reference [19]. These problems are used to test the accuracy and verify the Parafish and AZTRAN codes. The full specification (detailed geometry, and materials with their corresponding neutron cross-sections) for all problems is provided from [19] and modeled in cartesian coordinates. A brief description of the three problems is presented below.

### *3.1. Model 1 (Small LWR Core)*

The first model consists of a small model of a Light Water Reactor, based on the Kyoto University Critical Assembly (KUCA) modeled with two energy groups with the following energy cutoff $g1 = $ (10 MeV to 0.682 eV), and $g2 = $ (0.682 eV to 0.1 µeV) [19]. The geometry is shown in Figure 4 as one-eighth of geometry due to the reactor symmetry, and consists of a 25 cm × 25 cm × 25 cm cube reactor. It comprises a core fuel region with 15 cm × 15 cm × 15 cm. Neighboring to the core is a 5 cm × 5 cm × 25 cm control rod region, and all these regions are surrounded by reflector material. The specification of the problem provides two different configurations. The first one corresponds to the case where the control rod is wholly removed, filling it with an empty region. In the second case, the control rod is now completely inserted.

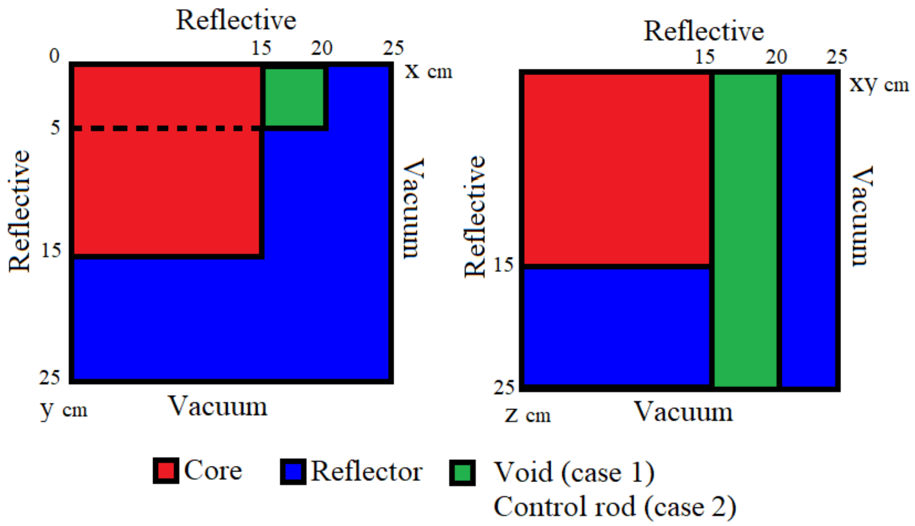

**Figure 4.** Core configuration of small LWR core.

### *3.2. Model 2 (Small FBR Core)*

The second model is based on a small core of a fast breeder reactor (FBR) and is modeled with four energy groups with the corresponding energy cutoff $g1 = $ (10 MeV to 1.353 MeV), $g2 = $ (1.353 MeV to 87.517 KeV), $g3 = $ (87.517 KeV to 0.961 KeV), and $g4 = $ (0.961 KeV to 0.1 µeV) [19]. The geometry model is shown in Figure 5 as a symmetric one-quarter reactor, and presents 70 cm × 70 cm × 150 cm dimensions. The model comprises a core region, radial and axial blanket, and control rod material. Similar to the previous model, two cases are presented. The first case consists of withdrawing the control rod entirely, and the region is filled with sodium (Na). In the second case, now just a half is filled with sodium (Na), and the other half of the control rod is inserted.

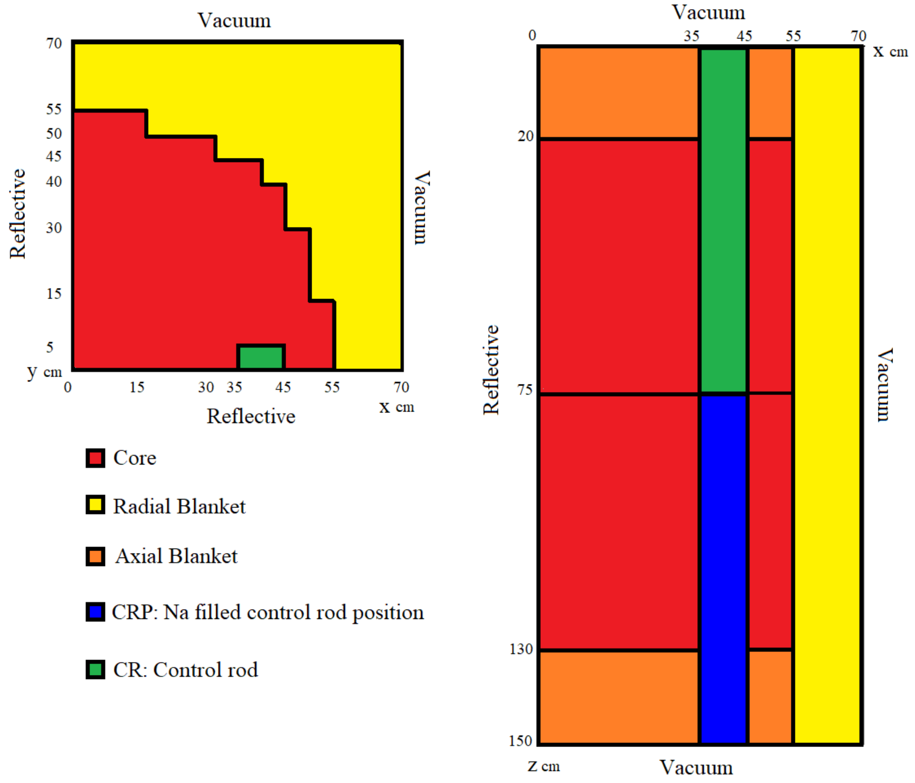

**Figure 5.** Core configuration of Small FBR core.

### 3.3. Model 3 (Axially Heterogeneous FBR Core)

Finally, the third model represents an axially heterogeneous FBR core also modeled with four energy groups like the previous model. Figure 6 shows the one-eighth symmetric part of the reactor with 160 cm × 160 × 90 cm of dimensions. The model is constituted by a core region, in addition to blanket regions (radial, axial and internal), a reflector region (radial and axial), and finally by an empty matrix and control rod regions, which makes it a very heterogeneous configuration. For this model, three different cases were calculated. In the first case, control rods are fully inserted (both center and off-center); in the second case, all control rods are entirely removed, and the regions are filled with sodium (Na). Finally, the control rods and sodium (Na) are replaced by blanket/core/reflector regions for the third case.

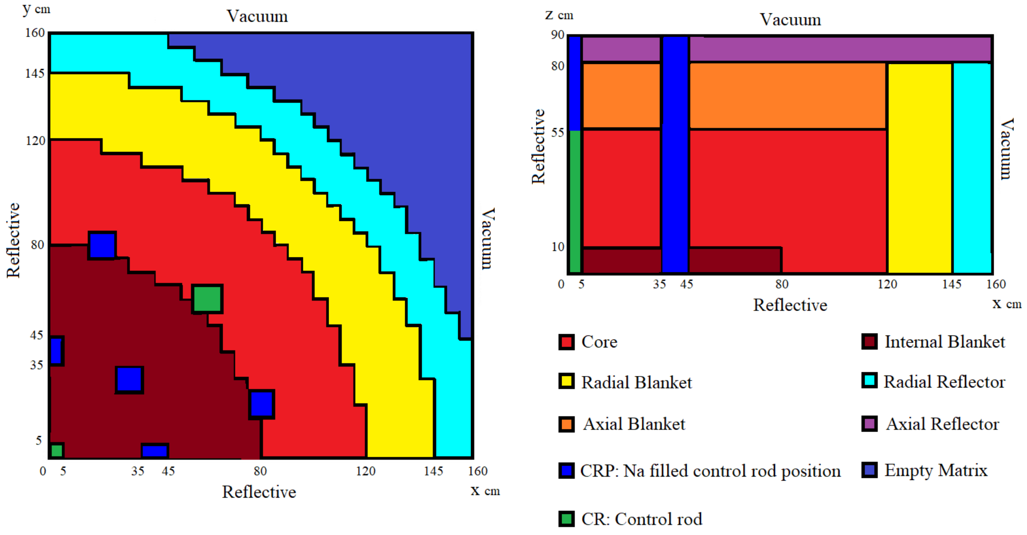

**Figure 6.** Core configuration of axially heterogeneous FBR core.

## 4. Numerical Results

Parafish and AZTRAN solutions to $k_{eff}$ value and the averaged scalar flux per region are presented and analyzed against the average Monte-Carlo calculation provided by [20]. To compare the discrepancy in $k_{eff}$ and region-averaged fluxes, the pcm relative error Equation (2), and relative difference Equation (3) are used, respectively.

$$\varepsilon_{pcm} = \frac{\left|(k_{eff}^{Ref} - k_{eff}^{Code})\right|}{k_{eff}^{Ref}} \times 10^5 \tag{2}$$

$$\varepsilon_{\%} = \frac{\left|(\overline{\phi}_{region}^{Ref} - \overline{\phi}_{region}^{Code})\right|}{\overline{\phi}_{region}^{Ref}} \times 100 \tag{3}$$

In addition, it is also considered that all the scalar fluxes $\phi$ have been normalized as Equation (4) and the control rod worth is calculated with Equation (5).

$$\sum_{g=1}^{G} \int_V \upsilon \Sigma_{f_g}(\vec{r}) \phi_g(\vec{r}) d\vec{r} = 1 \tag{4}$$

$$CR_{worth} = \frac{1}{k_{eff}^{in}} - \frac{1}{k_{eff}^{out}} \tag{5}$$

Finally, the convergence criteria applied for both codes are $10^{-6}$ for the neutron flux and $10^{-5}$ for $k_{eff}$. All cases were executed at the highest accuracy allowed by the workstation capabilities (Intel® Xeon® Processor E5-2697 v2 @2.70 GHz and 100 GB RAM).

### *4.1. Small LWR Core*

This calculation was modeled with a reference mesh size of 1 cm × 1 cm × 1 cm. Table 1 provides $k_{eff}$ values for both cases and the control rod reactivity worth. As can be observed, the results agree very well with the reference values, with differences of 22 pcm and 37 pcm errors for AZTRAN, while in the case of Parafish, the errors are 96 pcm and 9 pcm.

**Table 1.** Comparison of $k_{eff}$ and control rod worth for small LWR core.

| Code | Case 1 | Case 2 | CR-Worth |
|---|---|---|---|
| Monte-Carlo | 0.97780 | 0.96240 | $1.64 \times 10^{-2}$ |
| Parafish ($P_{11}$) | 0.97686 | 0.96249 | $1.52 \times 10^{-2}$ |
| | [96 pcm] | [9 pcm] | [7.3%] |
| AZTRAN ($S_{16}$) | 0.97758 | 0.96276 | $1.57 \times 10^{-2}$ |
| | [22 pcm] | [37 pcm] | [4.2%] |

In Figures 7 and 8, Parafish results are presented for the region-averaged flux per each energy group. The higher difference is found in the void/control rod region (1.03%/1.18%) associated with the thermal energy group. Regarding the other regions, the values are below 0.79% for case 1 and 0.31% for case 2. Figures 9 and 10 correspond to the same results using AZTRAN code. In this case, better precision is shown with errors less than 0.86% for the control rod/void region. In the other regions, the errors are less than 0.76% for case 1 and 0.12% for case 2. Both codes presented satisfactory results and similar behavior since the most important differences are found in the control rod/void region, due to the two group formulation and the strong difference in the nuclear properties of the materials in the control rod/void region.

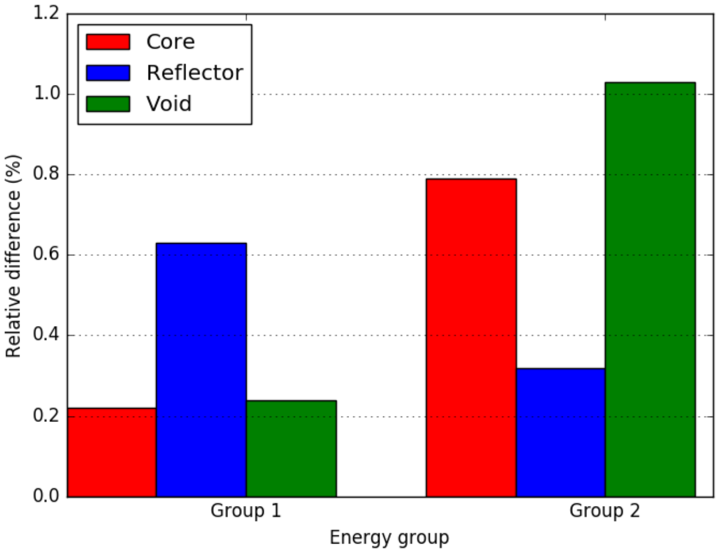

**Figure 7.** Comparison of region-averaged flux from reference and Parafish Model 1 Case 1.

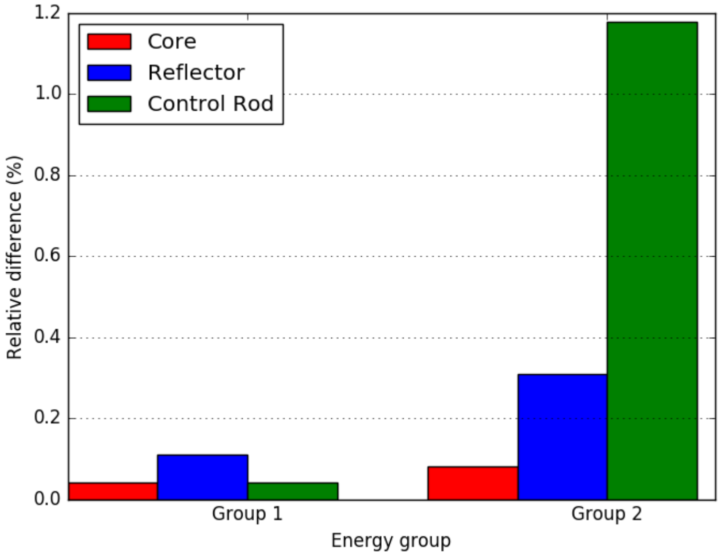

**Figure 8.** Comparison of region-averaged flux from reference and Parafish Model 1 Case 2.

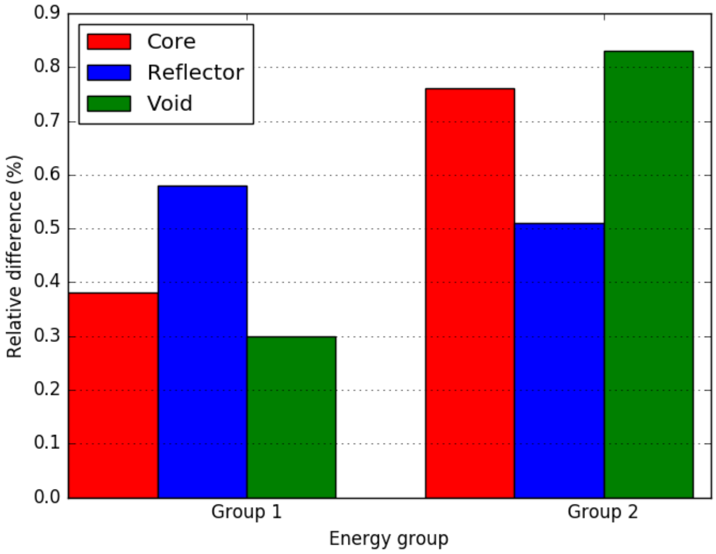

**Figure 9.** Comparison of region-averaged flux from reference and AZTRAN Model 1 Case 1.

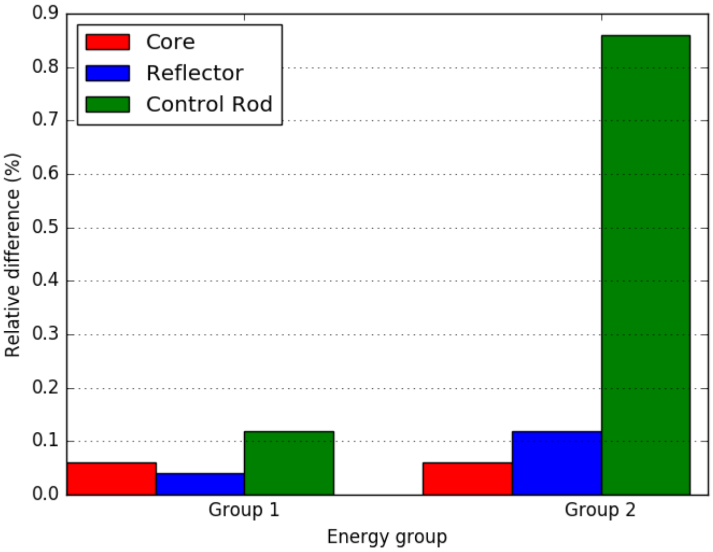

**Figure 10.** Comparison of region-averaged flux from reference and AZTRAN Model 1 Case 2.

### *4.2. Small FBR Core*

The problem is modeled with a reference mesh of 5 cm × 5 cm × 5 cm due to its large size. Table 2 supplies the $k_{eff}$ values obtained for each case, including the associated control rod worth. It can be seen that the values are in good agreement with the reference ones; the difference is lower than 263 pcm and 197 pcm when using Parafish and AZTRAN, respectively.

**Table 2.** Comparison of $k_{eff}$ and control rod worth for Small FBR core.

| Code | Case 1 | Case 2 | CR-Worth |
|---|---|---|---|
| Monte-Carlo | 0.97310 | 0.95890 | $1.48 \times 10^{-2}$ |
| Parafish $(P_{11})$ | 0.97511 | 0.96143 | $1.45 \times 10^{-2}$ |
| | [206 pcm] | [263 pcm] | [2.0%] |
| AZTRAN $(S_{16})$ | 0.97460 | 0.96079 | $1.47 \times 10^{-2}$ |
| | [154 pcm] | [197 pcm] | [0.6%] |

Figures 11 and 12 show significant errors in the relative differences obtained by Parafish (around 15%), especially in the sodium and control rod regions. Regarding the other regions, the radial blanket also presents a significant difference (8.75% case 1 and 11.91% case 2); however, the core and axial blanket results are quite good. One behavior that can be noticed is that the highest errors in the sodium and control rod regions are found in the fast group. In the case of the radial blanket, it is located in the thermal group. These considerable differences are far from the desired results; nevertheless, in the reference [19], Fletcher's results with MARK-PN code [21] presents similar phenomena, even with major differences. Therefore, it will be deepened if it is required to use a greater spatial-angular resolution or some technique to reduce these differences. On the other hand, Figures 13 and 14 provide the differences in region-averaged scalar fluxes achieved by AZTRAN. It can be observed that the relative differences are quite low for case 1, with higher values in sodium (0.85%) and axial blanket (0.79%), the differences in the other regions remain less than 0.3%. On the other hand, in case 2, a significant difference is found in the thermal group in the sodium (1.80%) and control rod (1.45%) regions. Despite that, the differences in the other regions are less than 0.55%. Overall, the agreement between AZTRAN calculation and reference values is satisfactory.

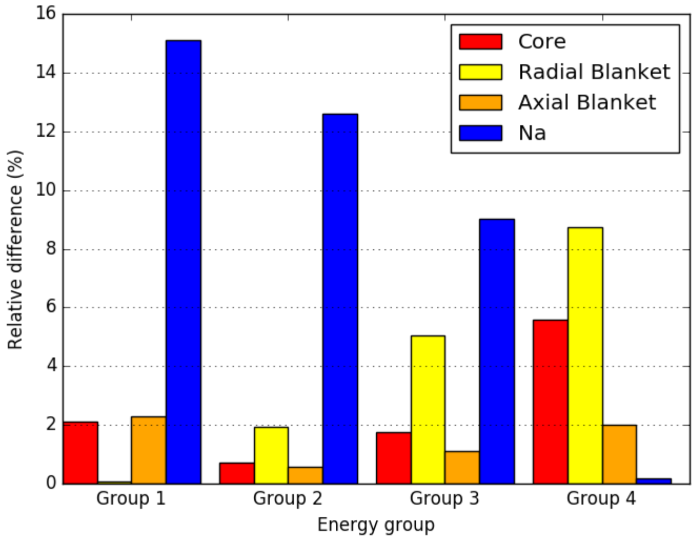

**Figure 11.** Comparison of region-averaged flux from reference and Parafish Model 2 Case 1.

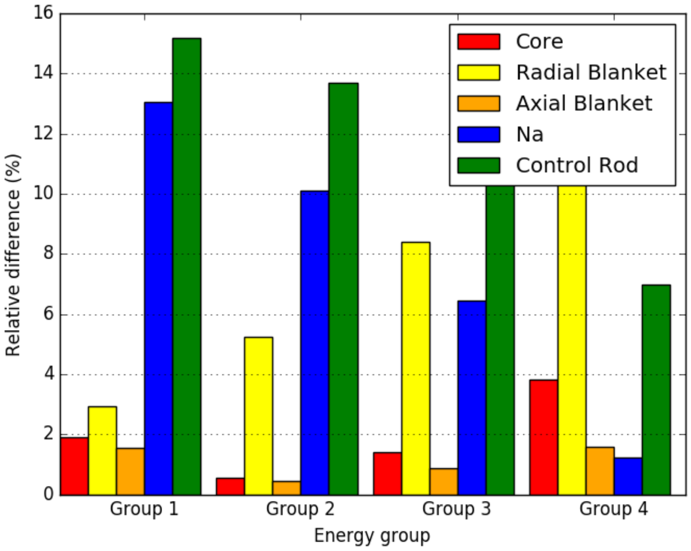

**Figure 12.** Comparison of region-averaged flux from reference and Parafish Model 2 Case 2.

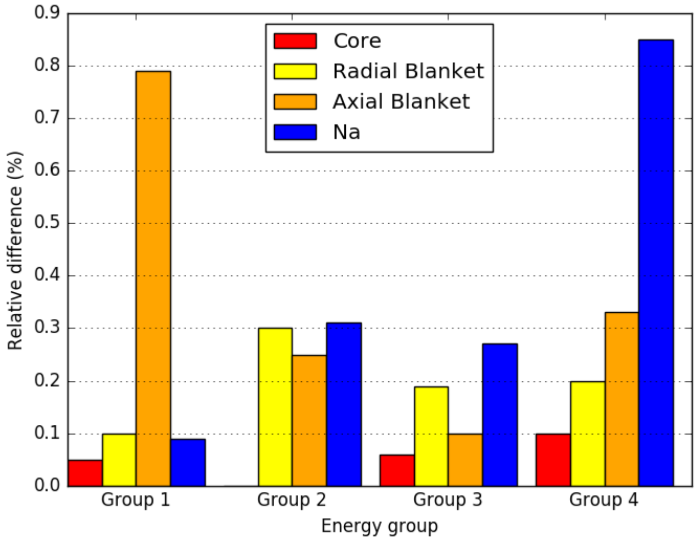

**Figure 13.** Comparison of region-averaged flux from reference and AZTRAN Model 2 Case 1.

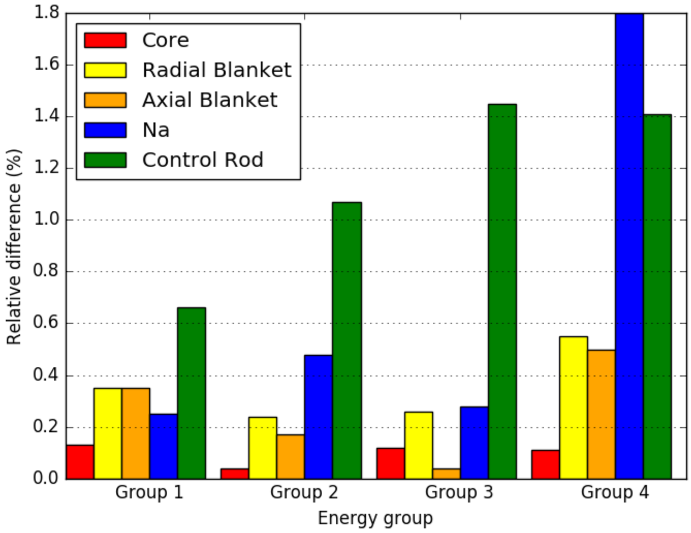

**Figure 14.** Comparison of region-averaged flux from reference and AZTRAN Model 2 Case 2.

### *4.3. Axially Heterogeneous FBR Core*

The cases were modeled with a reference mesh size of 5 cm × 5 cm × 5 cm. Due to lack of memory, Parafish had to use a more minor angular approximation than previously used ($P_5$). Table 3 lists $k_{eff}$ values for all obtained cases and their respective control rod worth. The Parafish results are in accordance with the reference ones, with differences being less than 267 pcm.

Concerning AZTRAN, cases 1 and 2 have a very good agreement with reference values within 179 pcm. However, a slightly larger difference of 393 pcm is found for case 3. In any case, these results indicate that Parafish and AZTRAN have acceptable accuracy in critical calculations for heterogeneous models.

**Table 3.** Comparison of $k_{eff}$ and control rod worth for Axially heterogeneous FBR core.

| Code | Case 1 | Case 2 | Case 3 | CR-Worth | CRP-Worth |
|---|---|---|---|---|---|
| Monte-Carlo | 0.97080 | 1.00050 | 1.02140 | $3.06 \times 10^{-2}$ | $2.03 \times 10^{-2}$ |
| Parafish ($P_5$) | 0.97340<br>[267 pcm] | 1.00230<br>[179 pcm] | 1.02290<br>[146 pcm] | $2.96 \times 10^{-2}$<br>[3.2%] | $2.01 \times 10^{-2}$<br>[0.9%] |
| AZTRAN ($S_{16}$) | 0.97254<br>[179 pcm] | 1.00175<br>[124 pcm] | 1.01738<br>[393 pcm] | $3.00 \times 10^{-2}$<br>[1.9%] | $1.53 \times 10^{-2}$<br>[24.6%] |

As far as the relative differences obtained by Parafish, Figures 15–17 show, once again, significant differences concerning the reference as the previous model, especially in Case 3, which is the most heterogeneous. Regarding the difference in regions, the sodium region is about 22.54% of case 1 and 19.73% of case 2; the other region with a notable difference is the radial blanket with errors around 8.3% case 1, 7.0% case 2, and 10.27% case 3, respectively. Regarding the different regions, results are below 2%, except for the internal blanket in case 3, which reaches a difference of up to 6.96%.

Figures 18–20 illustrate the relative differences related to AZTRAN. Most of the differences are below 1%, except in the control rod region that reaches 2% in the thermal group. Additionally, in case 3, the axial blanket has errors around 1.26–1.54%. Still, the results are satisfactory for this axially heterogeneous FBR and agree well with the reference values.

In both cases, differences are expected to be decreased by mesh refinement and angular approximation increased. Further investigations must be done in more robust workstations or clusters with enough memory.

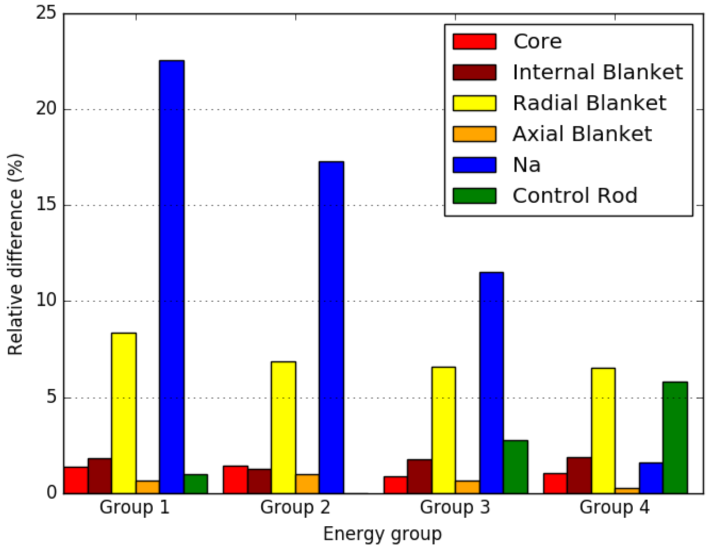

**Figure 15.** Comparison of region-averaged flux from reference and Parafish Model 3 Case 1.

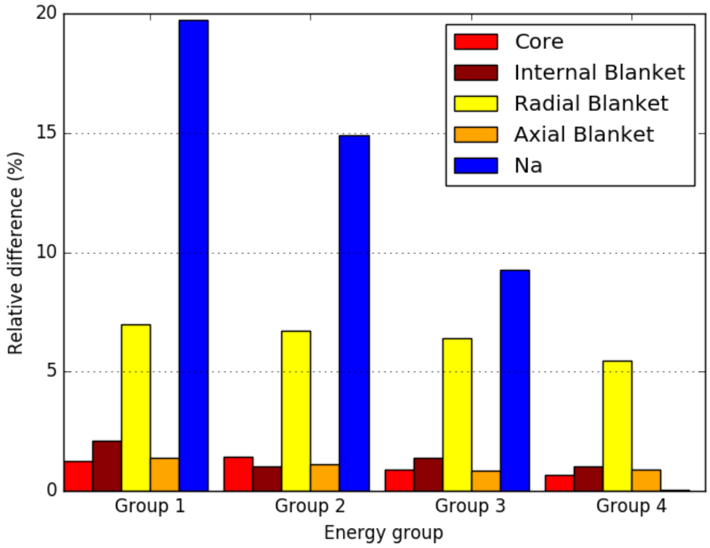

**Figure 16.** Comparison of region-averaged flux from reference and Parafish Model 3 Case 2.

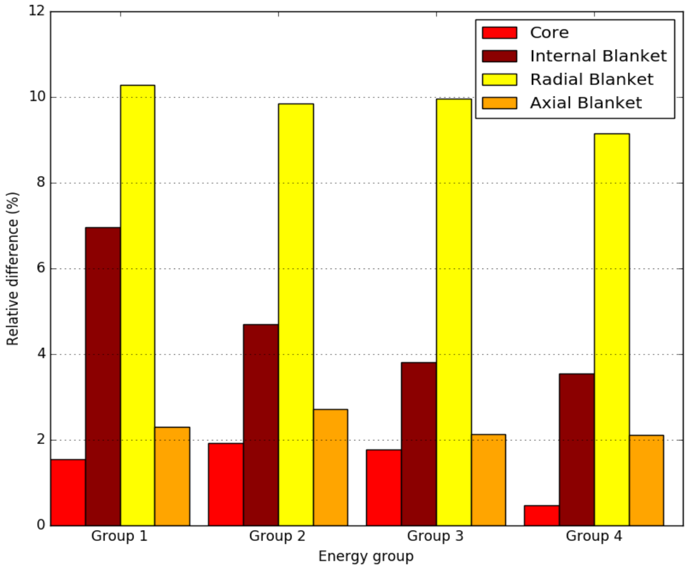

**Figure 17.** Comparison of region-averaged flux from reference and Parafish Model 3 Case 3.

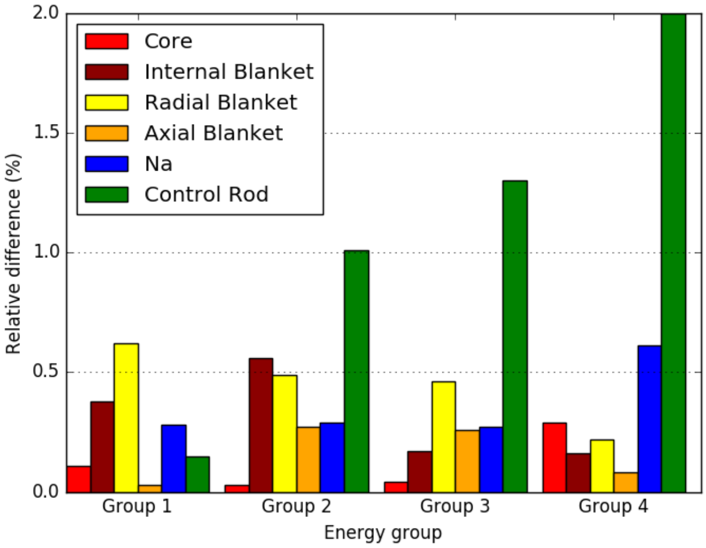

**Figure 18.** Comparison of region-averaged flux from reference and AZTRAN Model 3 Case 1.

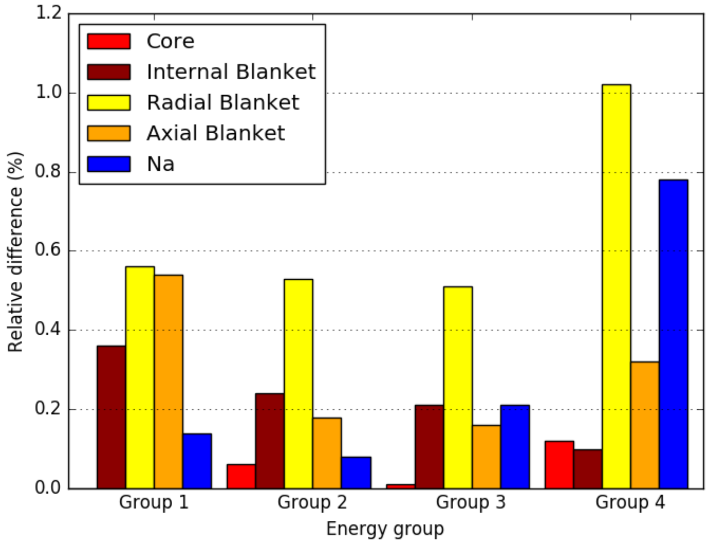

**Figure 19.** Comparison of region-averaged flux from reference and AZTRAN Model 3 Case 2.

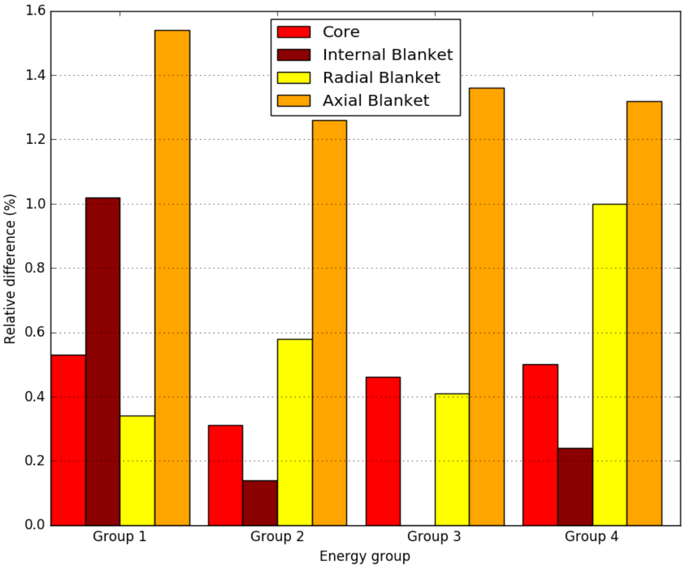

**Figure 20.** Comparison of region-averaged flux from reference and AZTRAN Model 3 Case 3.

## 5. Parallel Scaling

The TAKEDA Benchmark small FBR core was used to study the parallel performance reached by Parafish and AZTRAN. The speedup is defined as a sequential $(t_s)$ and parallel $(t_p)$ execution time function to quantify the parallel computation Equation (6).

$$speedup = \frac{t_s}{t_p} \tag{6}$$

The parallel speedup is shown in Figure 21. As shown, an acceptable speedup is reached by both codes, this being almost linear when using three processors. When further increasing the number of processors, a decrease in the speedup can be noticed due to the increase in the number of subdomains, which increases the communication between processors. It is notorious that Parafish has a drop in parallel scaling, and this may be since the increase in the subdomains leads to an increased duplication in the interface. In any case, a speedup factor of about five with eight processors is accomplished. AZTRAN's decrease is not so prominent, even though the inner iterations increase by increasing the subdomains. In this case, AZTRAN achieves a speedup of nearly seven with eight processors, which is an excellent scaling.

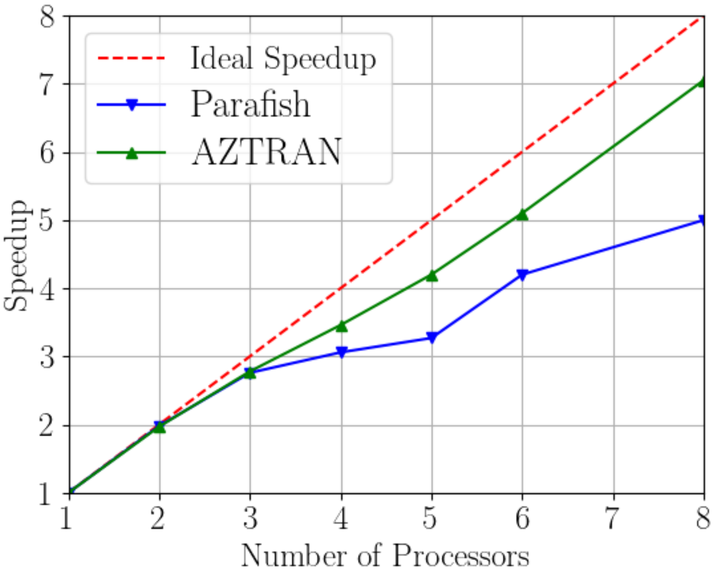

**Figure 21.** Speedup achieved by Parafish and AZTRAN for the small FBR core.

## 6. Conclusions and Outlook

The purpose of this work was to demonstrate the capabilities of Parafish and AZTRAN to solve 3D critical steady-state calculations. The well-known Takeda Benchmarks were used to test the codes under development and compare the results obtained against Monte Carlo reference solutions. In general, the two codes were found to provide $k_{eff}$ values within a good agreement concerning the reference ones.

Parafish has shown an excellent agreement only in the Model 1 cases as far as the region-averaged scalar fluxes. In contrast, some relatively significant differences can be highlighted for the other benchmark test cases, especially in cases where sodium is found. Therefore, more research needs to be done to mitigate these differences, such as increasing spatial and angular refinement on a more robust workstation. On the other hand, AZTRAN results are in excellent agreement with the reference ones since the discrepancies for all test cases are within 2%. As far as parallel scaling is concerned, both codes can be considered quite satisfactory. However, in the case of Parafish, the drop in parallel scaling as the number of processors increases is more noticeable than AZTRAN.

Given all these results, it could be said that AZTRAN has an advantage against Parafish; however, the development of AZTRAN is relatively new and is in constant development. Instead, Parafish is a development that has not been updated for ten years, and recently there has been an interest in improving the code. With the results obtained by this work, it was possible to recognize the deficiencies of the code to improve it in the immediate future. As for future work in AZTRAN, it has been identified that implementing the Diffusion Synthetic Acceleration (DSA) will reduce the inner iterations and improve the parallel scaling. In addition, a version with hybrid MPI (spatial) and OpenMP (energy) domain decomposition is under development, which will allow the exploitation of a greater amount of computational resources. In the Parafish case, the code requires several improvements with which it is intended to solve detailed problems in a reasonable computing time. Thus, the major update will be implementing an efficient parallel solver and probably considering an $SP_N$ approximation.

**Author Contributions:** Conceptualization, J.D.-G. and V.H.S.-E.; methodology, J.D.-G.; software, J.D.-G. and L.M.; validation, A.G.-T., E.d.V.-G. and L.M.; formal analysis, J.D.-G., A.G.-T. and E.d.V.-G.; investigation, J.D.-G.; resources, J.D.-G.; data curation, J.D.-G.; writing—original draft preparation, J.D.-G.; writing—review and editing, A.G.-T., E.d.V.-G., L.M. and V.H.S.-E.; visualization, J.D.-G.; supervision, V.H.S.-E.; project administration, V.H.S.-E.; funding acquisition, V.H.S.-E. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors would like to thank the HGF Program NUSAFE at KIT for the financial support. In addition, the authors acknowledge the financial support from the National Strategic Project No. 212602 (AZTLAN Platform) as part of the Sectorial Fund for Energetic Sustainability CONACYT–SENER (Mexico). Finally, we acknowledge support by the KIT-Publication Fund of the Karlsruhe Institute of Technology.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Fletcher, J.K. The Solution of the Multigroup Neutron Transport Equation Using Spherical Harmonics. *Nucl. Sci. Eng.* **1983**, *84*, 33–46. [CrossRef]
2. Lewis, E.E.; Miller, W.F. *Computational Methods of Neutron Transport*; Wiley: Hoboken, NJ, USA, 1984.
3. Lathrop, K.D. Ray effects in discrete ordinates equations. *Nucl. Sci. Eng.* **1968**, *32*, 357–369. [CrossRef]
4. Morel, J.; Wareing, T.; Lowrie, R.; Parsons D. Analysis of ray-effect mitigation techniques. *Nucl. Sci. Eng.* **2003**, *144*, 1–22. [CrossRef]
5. Smith, M.A.; Lewis, E.E.; Shemon, E.R. *DIF3D-VARIANT 11.0: A Decade of Updates*; Argonne National Lab. (ANL): Argonne, IL, USA , 2014. [CrossRef]
6. Ziver, A.K.; Shahdatullah, M.S.; Eaton, M.D.; Oliveira, C.R.E.; Umpleby, A.P.; Pain, C.C.; Goddard, A.J.H. Finite element spherical harmonics (PN) solutions of the three-dimensional Takeda benchmark problems. *Ann. Nucl. Energy* **2005**, *32*, 925–948. [CrossRef]
7. Alcouffe, R.E.; Baker, R.S.; Dahl, J.A.; Turner, S.A.; Ward, R. *PARTISN: A Time-Dependent, Parallel Neutral Particle Transport Code System;* LA-UR-08-07258. Los Alamos National Laboratory: Los Alamos, NM, USA, 2008.
8. Seubert, A.; Velkov, K.; Langenbuch, S. The time-dependent 3-d discrete ordinates code TORT-TD with thermal-hydraulic feedback by ATHLET models. In Proceedings of the Physor 2008, Interlaken, Switzerland, 14–19 September 2008.
9. Criekingen, S.V.; Nataf, F.; Havé, P. PARAFISH: A parallel FE–PN neutron transport solver based on domain decomposition. *Ann. Nucl. Energy* **2011**, *38*, 145–150. [CrossRef]
10. Criekingen, S.V. A Non-Conforming Generalization of Raviart-Thomas Elements to the Spherical Harmonic form of the Even-parity Neutron Transport Equation. *Ann. Nucl. Energy* **2006**, *33*, 573–582. [CrossRef]
11. Subramanian, C.; Criekingen, S.V.; Heuveline, V.; Nataf, F.; Havé, P. The Davidson method as an alternative to power iterations for criticality calculations. *Ann. Nucl. Energy* **2011**, *38*, 2818–2823. [CrossRef]
12. Nakamura, S. *Computational Methods in Engineering and Science: With Applications to Fluid Dynamics and Nuclear Systems*; Wiley: New York, NY, USA, 1977.

13. Saad, Y.; Schultz, M.H. GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM J. Sci. Statis. Comput.* **1986**, *7*, 856–869. [CrossRef]
14. Lions, P. L. On the Schwarz Alternating Method III: A variant for Nonoverlapping Subdomains. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*; SIAM: Philadelphia, PA, USA, 1990; pp. 202–223.
15. Gomez-Torres, A.; del Valle-Gallegos, E.; Duran-Gonzalez, J.; Rodriguez-Hernandez, A. Recent developments in the neutronics codes of the AZTLAN platform. In Proceedings of the International Congress on Advances in Nuclear Power Plants ICAPP, Abu Dhabi, United Arab Emirates, 16–20 October 2021.
16. Duran-Gonzalez, J.; del Valle-Gallegos, E.; Reyes-Fuentes, M.; Gomez-Torres, A.; Xolocostli-Munguia, V. Development, verification, and validation of the parallel transport code AZTRAN. *Prog. Nucl. Energy* **2021**, *137*, 103792. [CrossRef]
17. Hennart, J.P.; del Valle, E. Nodal finite element approximations for the neutron transport equation. *Math. Comput. Simul.* **2010**, *80*, 2168–2176. [CrossRef]
18. Yavu, M.; Larsen E.W. Iterative Methods for Solving x-y Geometry SN Problems on Parallel Architecture Computers. *Nucl. Sci. Eng.* **1992**, *112*, 32–42. [CrossRef]
19. Takeda, T.; Ikeda, H. 3D neutron transport benchmarks. Technical Report OECD/NEA, Committee on Reactor Physics (NEACRP), OSAKA University, NEACRP-L-330. 1991. Available online: https://www.oecd-nea.org/upload/docs/application/pdf/2020-01/neacrp-l-1990-330.pdf (accessed on 24 December 2021).
20. Takeda, T.; Ikeda, H. 3D neutron transport benchmarks. *Nucl. Sci. Technol.* **1991**, *28*, 656–669. [CrossRef]
21. Fletcher, J.K. *MARK/PN A Computer Program to Solve the Multigroup Transport Equation*; RTS-R-002; AEA: Risley, UK, 1988 .