Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 5782146, 9 pages
https://doi.org/10.1155/2019/5782146

Research Article

Richardson Extrapolation-Based Verification Method of Scientific Calculation Program without the Oracles: A Case Study

Shiyu Yan,[1,2] Xiaohua Yang,[1,2] Guodong Cheng,[1,2] and Hua Liu [2]

[1]*Computer School, University of South China, Hengyang, Hunan Province 421001, China*
[2]*CNNC Key Laboratory on High Trusted Computing, University of South China, Hengyang, Hunan Province 421001, China*

Correspondence should be addressed to Hua Liu; lhsmile@usc.edu.cn

Received 1 October 2018; Accepted 2 December 2018; Published 30 January 2019

Academic Editor: Fausto Arpino

For the verification test of some scientific calculation programs, different comparison methods are commonly applied to ensure the correctness of the computations. However, it is difficult to verify whether the testing output is correct, because the oracles which include the expected output are not always available or too hard to get. For this reason, the authors focus on using the Richardson Extrapolation to estimate the convergences of the numerical solution on different levels of mesh refinement. These numerical convergence properties can be applied to verification test, without the need for giving the oracles. In the present study, the authors take the program test of the multigroup neutron diffusion equations as a study case and propose the Richardson Extrapolation-based verification method. Three verification criterions are obtained based on our approach. In addition, a test experiment is conducted demonstrating the validity of our theoretical results.

1. Introduction

The importance of scientific software in mission-critical and safety-critical applications has increased dramatically during the last three decades. The engineers, developers, and users of scientific software face a critical problem: How should confidence in computational science and engineering be critically assessed [1]? Since any subtle defects in critical system software will cause catastrophic consequences, the credibility of the computational results must be raised to a higher level.

Verification test is the major process for improving this confidence of the software quality. Great progress has been made in the aspects of verification in scientific software [2]. There remains a great challenge in this verification community; that is, there is no reliable test oracle to indicate what the correct output should be for arbitrary input, which was called oracle problem. This problem commonly arises when conducting verification testing of scientific software; many scientific applications fall into the category of nontestable programs [3] where an oracle is unavailable or too difficult to implement. However, many testers of scientific software generally adopt the following mechanisms as test oracles: (a) comparing with analytical solutions, simulation results, tabulated values, or hand-calculations [4]; (b) verifying with standard mathematical libraries or reference software packages [5]. Additionally, other methods for solving the oracle problem have been developed, including formal proof [6], analytic solutions for simplified physics, method of manufactured solutions [7], PDE benchmark solutions, and metamorphic testing [8, 9]. For example, around 30 verification benchmarks have been developed by the National Agency for Finite Element Methods and Standards (NAFEMS), which have been used widely in verification. In addition, ANSYS [10] and ABAQUS have roughly 270 formal verification tests. Readers are referred to [11–13] for more details on verification. The commonality of these methods is to use the analytical solution, simulation results, and reference values as test oracle. However these methods are unavailable for every newly investigated program.

In order to solve the oracle problem in scientific applications, the mathematical models and numerical methods are factors that should be investigated again. The a posteriori estimation methods based on Richardson Extrapolation are examined [15]. Emphasis is placed on discretization error estimation methods based on Richardson Extrapolation [16]. In the process of using Richardson Extrapolation to analyze the order of error accuracy of the numerical solution, the exact solution of the partial differential equation is actually an auxiliary role and does not need to be given. In other words, Richard Extrapolation methods can be applied to verify the no-test oracle problem. Moreover, the relevant verification criterions could be constructed by the analysis process based on Richardson Extrapolation method [16].

Since the quality of the nuclear reactor design software is closely related to the safety and economy of reactors, the verification test is very important for improving the credibility of programs. Actually, the program for solving multigroup diffusion equations is the primary procedure of the nuclear reactor design software [14]. In the critical state of the reactor, the multigroup diffusion equations are usually described as the solution of Lambda Modes problem or λ-eigenvalue problem [17], which provides the fundamental eigenvalue (the maximal eigenvalue) that is called the effective multiplication factor (k-effective). In this case, the neutron flux is a corresponding eigenfunction. Computations of the Lambda Modes problem provide the results of the k-effective and neutron flux distribution for reactor design calculations. Hence, if the program is calculated incorrectly, it will lead to error prediction and affect the safety and economy of the reactor. Though traditional comparison methods are commonly applied in verification test of partial differential equations program, it is difficult to give expected values to verify whether the calculation results are correct.

In the present study, the program of the multigroup neutron diffusion equations is performed as a study case. We investigate the mathematical models and the finite difference numerical algorithm of the multigroup diffusion equations. Based on Richardson Extrapolation, the convergence properties of numerical solution in H^0 and H^1 are obtained, and then the rigorous verification criterions are constructed for verification. However, we would not need to give exact solutions to verification test.

The rest of the paper is organized as follows. In Section 2 we introduce the mathematical model. Section 3 presents the main numerical method and convergence analysis process based Richardson Extrapolation. Test experiments are conducted, and the results are reported in Section 4. Finally, we give a summary and conclude the paper in Section 5.

2. Background

2.1. Mathematical Model. The steady state multigroup neutron diffusion equations are the basic mathematical model in reactor design, where the governing equations are given by (1) and (2).

$$-\nabla \cdot D_g(r)\nabla\phi_g(r) + \Sigma_{R,g}(r)\phi_g(r) = \sum_{g'=1,g'\neq g}^{G} \Sigma_{g'\to g}(r)\phi_{g'}(r) + \frac{\chi_g}{k_{eff}}Q(r), \quad g = 1,2,\ldots,G, \tag{1}$$

$$Q(r) = \sum_{g'=1}^{G} \left(\upsilon\Sigma_f(r)\right)_{g'}\phi_{g'}(r), \tag{2}$$

where g is the energy group, $D_g(r)$ is the diffusion coefficient of the energy group g, $\Sigma_{R,g}(r)$ is the removal cross section of the energy group g, $\Sigma_{g'\to g}(r)$ is the scattering cross section from energy group g' to g, υ_g is the average number of neutrons emitted by fission of the energy group g, $(\sum_f(r))_{g'}$ is the fission cross section of the energy group g', $\phi_g(r)$ is the neutron scalar flux of the energy group g, χ_g is the integrated fission spectrum of the energy group g, and k_{eff} is the effective multiplication factor of reactor. $Q(r)$ is the fission term. In the multigroup formulation, the neutron diffusion equations are represented by a coupled system of partial differential equations for the flux.

Assuming a bounded 2D or 3D domain Ω with a convex boundary $\partial\Omega$, the domain Ω is composed of the finite number of subassemblies Ω_i, $i = 1,2,\ldots,M$. For $r \in \Omega$ with an outer boundary conditions, we have

$$D_g\frac{\partial\phi_g(r)}{\partial n} + \phi_g(r) = 0, \quad r \in \partial\Omega, \tag{3}$$

where n is the outer normal to the boundary $\partial\Omega$; at an interface between dissimilar regions Ω_i with continuous conditions, we have

$$\phi_g(r_+) = \phi_g(r_-), \tag{4}$$

and

$$D_g\frac{\partial\phi_g(r_+)}{\partial n} = D_g\frac{\partial\phi_g(r_-)}{\partial n}, \quad r \in \partial\Omega_i, \tag{5}$$

where r_+ and r_- represent the value of both sides of the interface, respectively. The quantities $\Sigma_{R,g}(r)$, $D_g(r)$, $\Sigma_{g'\to g}(r)$, χ_g, $\upsilon\Sigma_f(r)$ are constant in each Ω_i.

To simplify the manipulations, (1) and (2) are expressed in a matrix system by the following form:

$$\mathbf{M}\boldsymbol{\varphi} = \frac{1}{k_{eff}}\mathbf{F}\boldsymbol{\varphi}, \tag{6}$$

where the vector of the neutron flux is $\boldsymbol{\varphi} = [\phi_1,\phi_2,\ldots,\phi_G]^T$, the coefficient matrix $\mathbf{M} = (a_{i,j})_{G,G}$ is defined with

$$a_{i,j} = \begin{cases} -D_i\nabla^2 + \Sigma_{R,i}, & i = j \\ -\Sigma_{j',i}, & i \neq j', \end{cases} \tag{7}$$

and the coefficient matrix $\mathbf{F} = (f_{i,j})_{G,G}$ is defined with $f_{i,j} = \chi_i\upsilon_j\Sigma_{fj}$.

2.2. Power Iteration. The power iteration (PI) method is the most used method to obtain the fundamental eigenvalue and its eigenvector [18]. This iterative source method is described in detail by [19]. The power iteration algorithm process is as follows.

(I) Starting with a positive initial estimate for $\boldsymbol{\varphi}^{(0)}$ and $k_{eff}^{(0)}$, they are substituted into the right hand of (6) to obtain the neutron flux $\boldsymbol{\varphi}^{(1)}$. The upper labels represent the number of iterations.

(II) For the i-th iteration, (6) is written as

$$\mathbf{M}\boldsymbol{\varphi}^{(i+1)} = \frac{1}{k_{eff}^{(i)}}\mathbf{S}^{(i)}, \tag{8}$$

where $\mathbf{S} = \mathbf{F}\boldsymbol{\varphi}$.

(III) Updating the value for new k_{eff} and the new flux vector $\boldsymbol{\varphi}$, $k_{eff}^{(i+1)}$ and $\mathbf{S}^{(i+1)}$ are calculated by the following equations, respectively,

$$\begin{aligned} \mathbf{S}^{(i+1)} &= \mathbf{F}\boldsymbol{\varphi}^{(i+1)}, \\ k_{eff}^{(i+1)} &= k_{eff}^{(i)} \frac{\int_\Omega \mathbf{S}^{(i+1)} d\Omega}{\int_\Omega \mathbf{S}^{(i)} d\Omega}, \end{aligned} \tag{9}$$

with the stop criterion as follows:

$$\begin{aligned} \frac{\left| k_{eff}^{(i+1)} - k_{eff}^{(i)} \right|}{\left| k_{eff}^{(i+1)} \right|} &< \varepsilon_1, \\ \frac{\left| \int_\Omega \mathbf{S}^{(i+1)} d\Omega - \int_\Omega \mathbf{S}^{(i)} d\Omega \right|}{\left| \int_\Omega \mathbf{S}^{(i+1)} d\Omega \right|} &< \varepsilon_2, \end{aligned} \tag{10}$$

where ε_1 and ε_2 are small constants.

(IV) If the result does not correspond to the stop criterion, return to step (II).

The iteration process of step (II) is called inner iteration, which usually applies the finite differences, finite elements, and other methods to solve (1). Readers are referred to [17, 20, 21] for more details on the mathematical theory. In particular, the finite difference method is a widely used method.

At each step of the power iteration procedure, if we omit the superscript and subscript, (1) is expressed as G uncoupled self-consistency elliptic boundary value problems. Equation (1) is rewritten by

$$-\nabla D(r) \nabla \phi(r) + \Sigma_R(r) \phi(r) = S(r), \quad r \in \Omega \tag{11}$$

$$D(r) \frac{\partial \phi(r)}{\partial n} + \phi(r) = 0, \quad r \in \partial\Omega. \tag{12}$$

Here, $S(r)$ contains the fission and scattering terms. Actually, the fission source is not known so that the inner iteration must be embedded in step (III), which is called outer iteration. Therefore, we fill the values of the inner iteration to the outer iteration.

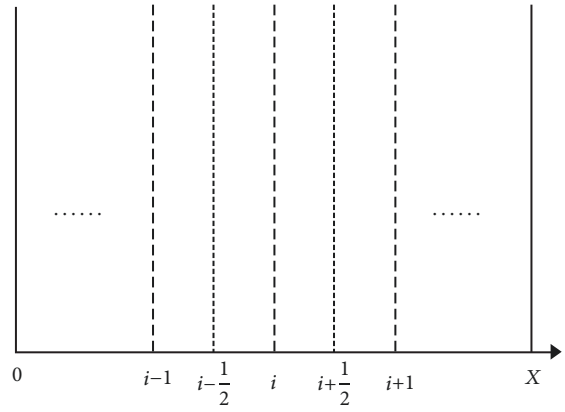

Figure 1: Space discrete in one-dimensional X geometry.

3. Scientific Software and Numerical Method

Many large and complex reactor physics calculation software packages include programs for solving the multigroup neutron diffusion equations, which widely adopted the differential method to carry on spatial discretization [22]. In the current study, we design and compile a calculation program called HARMONY-2, which also adopts the finite difference method to discrete the multigroup neutron diffusion equation, and invokes ARPACK which is an open source package developed by the Department of Computing and Applied Mathematics of RICE University. The ARPACK implements the IRAM (Implicitly Restarted Arnoldi Method) algorithm, which is presented in [23]. HARMONY-2 is used to calculate high order harmonics of the neutron diffusion equations.

3.1. The Finite Difference Method. We consider an example of one-dimensional and multigroup neutron diffusion equations:

$$\begin{aligned} -\frac{d}{dr}\left(D_g(r) \frac{d\phi_g}{dr} \right) + \Sigma_{R,g}(r) \phi_g(r) &= S_g(r), \\ r \in \Omega \ g = 1, 2, \ldots, G, \end{aligned} \tag{13}$$

where $\Omega = \{r : 0 < r < R\}$, $\partial\Omega = \{0, R\}$, ϕ_g in $\partial\Omega$ is zero, and ϕ_g is continuous inside of Ω.

Assuming Ω is divided into a limited number of subregions. Space discretization in one-dimensional X geometry is shown in Figure 1.

Let Δx be the split size and NX be the split number in the X geometry. The medium in each subdivision area is uniform, so the various cross sections and diffusion coefficients are constant. Using a backward difference format, we obtain the equations as formulation (14).

$$\begin{aligned} &a_g(i) \phi_{g,m}(i-1) + b_g(i) \phi_{g,m}(i) + c_g(i) \phi_{g,m}(i+1) \\ &\quad - SS_{g,m}(i) = FS_{g,m}(i) \\ &\qquad g = 1, 2, \ldots, G, \ i = 1, 2, \ldots, NX \end{aligned} \tag{14}$$

Equation (14) is written as an overrelaxation iterative format:

$$\begin{aligned}\phi_{g,k}^{(K)} = {} & \frac{\omega}{b_g(i)}\left[FS_{g,m-1}(i)\right] + SS_{g,m}^{(K-1)}(i) \\ & - a_g(i)\,\phi_{g,m}^{(K)}(i-1) - c_g(i)\,\phi_{g,m}^{(K-1)}(i+1) \\ & + (1-\omega)\,\phi_{g,m}^{(K-1)}(i),\end{aligned} \tag{15}$$

where superscript K is the inner iteration count and m is the outer iteration count of calculation algorithm, where

$$a_g(i) = -\frac{D_g(i-1/2)}{\Delta x(i-1)}, \tag{16}$$

$$\begin{aligned}b(i) = {} & \frac{1}{2}\Sigma_{t,g}\left(i+\frac{1}{2}\right)\Delta x(i) \\ & + \frac{1}{2}\Sigma_{t,g}\left(i-\frac{1}{2}\right)\Delta x(i-1) - a_g(i) \\ & - c_g(i),\end{aligned} \tag{17}$$

$$c_g(i) = -\frac{D_g(i+1/2)}{\Delta x(i)}, \tag{18}$$

$$FS_g(i) = \chi_g(i)\sum_{g'=1}^{G}\left(\upsilon\Sigma_f\right)_{g'}\phi_{g',m-1}(i)\,\Delta x(i), \tag{19}$$

$$SS_{g,m}(i) = \sum_{g'=1}^{G}\Sigma_{g'\longrightarrow g}(i)\,\phi_{g',m}(i)\,\Delta x. \tag{20}$$

Boundary conditions are as follows:

$$\begin{aligned}a_g\left(i_B^-\right) &= -\frac{2D_g\left(i_B^-\right)}{\Delta x\left(i_B^-\right)}, \\ \phi_g\left(i_B^- - 1\right) &= 0,\end{aligned} \tag{21}$$

$$\begin{aligned}a_g\left(i_B^+\right) &= -\frac{2D_g\left(i_B^+\right)}{\Delta x\left(i_B^+\right)}, \\ \phi_g\left(i_B^+ + 1\right) &= 0.\end{aligned} \tag{22}$$

Calculation flow of the program algorithm for (13) is shown in Figure 2.

3.2. Richardson Extrapolation. Based on the Richardson Extrapolation method, the convergence of the numerical solution in (1) and (2) was investigated. Let φ_h be the numerical solution of (13) in a suitably fine mesh and φ be the exact solution of this problem. The Richardson Extrapolation procedure assumes that the numerical scheme is first-order accurate or second-order accurate. Considering the Sobolev space $H^\alpha(\Omega)$, we assume $\varphi_h \in H^\alpha(\Omega)$, and we have

$$\left\|\varphi - \varphi_h\right\|_{H^\alpha(\Omega)} = o\left(h^{2-\alpha}\right). \tag{23}$$

Specifically, the numerical solution φ_h is calculated by the same discretization scheme with three different grid sizes: $h_1 = h$, $h_2 = h/2$, and $h_3 = h/4$, respectively.

If $\alpha = 1$, the error can be rewritten by

$$\varphi_h = \varphi_{exact} + c_0 h + o\left(h^2\right), \tag{24}$$

$$\varphi_{h/2} = \varphi_{exact} + \frac{c_0 h}{2} + o\left(\left(\frac{h}{2}\right)^2\right), \tag{25}$$

$$\varphi_{h/4} = \varphi_{exact} + \frac{c_0 h}{4} + o\left(\left(\frac{h}{4}\right)^2\right). \tag{26}$$

Equations (24) to (26) yield

$$\left\|\varphi_h - \varphi_{h/2}\right\|_{H^1} = \frac{c_0 h}{2}, \tag{27}$$

$$\left\|\varphi_{h/2} - \varphi_{h/4}\right\|_{H^1} = \frac{c_0 h}{4}. \tag{28}$$

Then, we get

$$\frac{\left\|\varphi_h - \varphi_{h/2}\right\|_{H^1}}{\left\|\varphi_{h/2} - \varphi_{h/4}\right\|_{H^1}} \cong 2. \tag{29}$$

If $\alpha = 0$, we have

$$\varphi_h = \varphi_{exact} + c_1 h^2 + o\left(h^3\right), \tag{30}$$

$$\varphi_{h/2} = \varphi_{exact} + c_1\left(\frac{h}{2}\right)^2 + o\left(\left(\frac{h}{2}\right)^3\right), \tag{31}$$

$$\varphi_{h/4} = \varphi_{exact} + c_1\left(\frac{h}{4}\right)^2 + o\left(\left(\frac{h}{4}\right)^3\right), \tag{32}$$

and that the mesh is refined or coarsened by a factor of two. Similarly, we have the following.

$$\frac{\left\|\varphi_h - \varphi_{h/2}\right\|_{H^0}}{\left\|\varphi_{h/2} - \varphi_{h/4}\right\|_{H^0}} \cong 4 \tag{33}$$

Based on the above derivation, the numerical solution convergence of H^0 and H^1 was obtained by Richardson Extrapolation method. During the above analysis process, it is very clear that the exact solutions of neutron diffusion equations are eliminated in the derivation, so it is not needed to provide the exact solution as the test oracle. Moreover, it is very convenient to verify the program by checking whether the numerical solution satisfies the convergence properties. Moreover, as the mesh is encrypted, the neutron flux and Max-eigenvalues (effective multipliers) converge, respectively, which is also a property that can be used for verification. Based on the convergence properties, three verification criterions are constructed. The first and second one are the numerical solution convergence of H^0 and H^1, respectively. The last one is the Max-eigenvalues convergence with the mesh refinement.

4. Test Experiments

4.1. SLAB_1D_2G Benchmark Problem. The purpose of the test experiment is to apply the three verification criterions

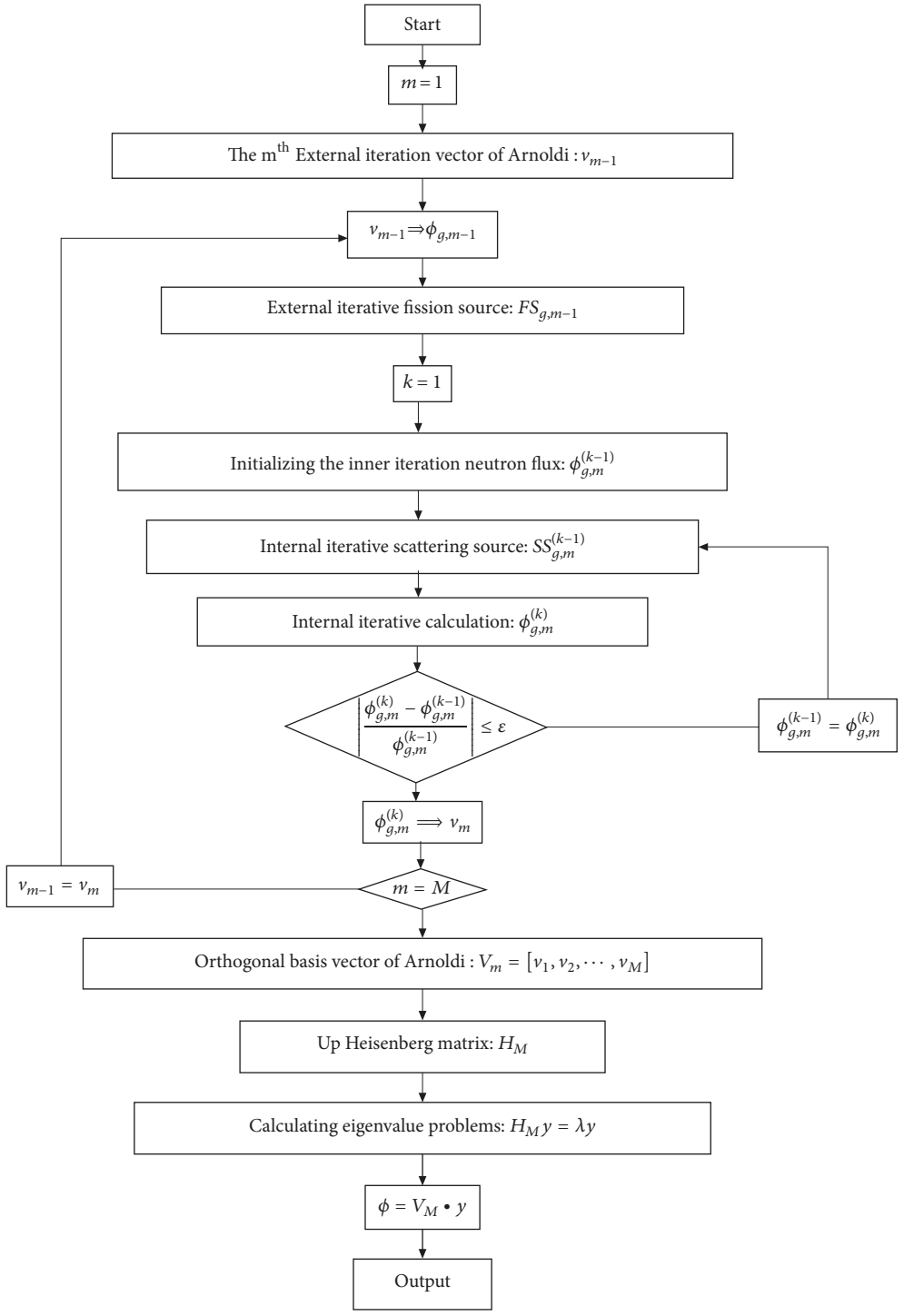

Figure 2: Calculation flow chart of the algorithm of HARMNOY-2.

presented above to verify the program HARMNOY-2. In this experiment, we select the SLAB_1D_2G benchmark problem as the computational case, which is the one-dimensional slab reactor and double-group diffusion problem [14]. The thickness of the slab is 450cm with uniform mesh. The reactor core geometry is shown in Figure 3, and the cross section parameters are shown in Table 1.

(1) The HARMONY-2 program performs the SLAB_1D_2G benchmark calculation. The grid is divided into 18 equal parts in one-dimensional Cartesian coordinate system. For comparison with [14], the output results of the eigenvalues are set as four modes eigenvalues.

(2) Verification testing by three verification criterions: Let neutron flux be $\varphi^h = (\phi_1^h, \phi_2^h)$, where ϕ_1^h is the

Table 1: Parameters of SLAB_1D_2G one-dimensional slab benchmark problem.

Energy group	D/cm	Σ_t/cm^{-1}	$\nu\Sigma_f/cm^{-1}$	χ_p/χ_d	$\Sigma_{1\to 2}/cm^{-1}$
1	1.264E-00	8.154E-03	0.000E-00	1.00E-00	7.368E-03
2	9.328E-01	4.100E-03	4.562E-03	0.00E-00	7.368E-03

Table 2: Comparison of results by the four modes eigenvalues.

n	λ-eigenvalue		Absolute error
	HARMONY-2	Lecture [14]	
1	9.86988E-01	9.8698E-01	0.0000
2	9.35159E-01	9.3516E-01	1.000E-05
3	8.58979E-01	8.5898E-01	1.000E-05
4	7.69751E-01	7.6976E-01	1.000E-05

Table 3: The neutron flux values of x_{mid} in different refined grids.

Mesh	Element	The neutron flux value of x_{mid} for 1th group energy.	*Ratio*1	The neutron flux value of x_{mid} for 2th group energy.	*Ratio*1
h_1	18	6.641297145594463E-002		6.641294098874310E-002	
h_2	36	6.660342485457200E-002		6.660331582277153E-002	
h_3	72	6.665076278267033E-002	4.02327280213385	6.665082494407264E-002	4.00712176556248
h_4	144	6.666264561430789E-002	3.98372454834075	6.666253389102599E-002	4.05750589616633
h_5	288	6.666565458515177E-002	3.94913485510786	6.666562905904372E-002	3.78297620234630
h_6	576	6.666633289577738E-002	4.43597775157830	6.666641383170335E-002	3.94403140799427

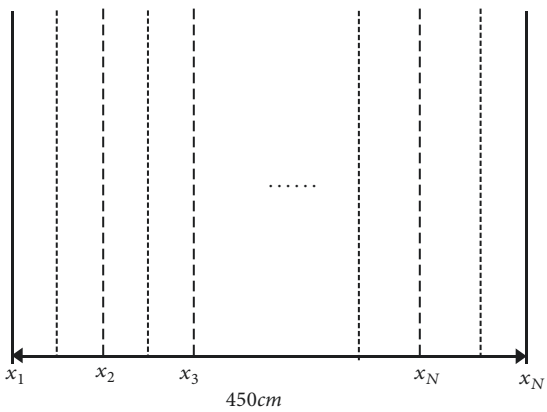

Figure 3: Reactor core configuration of SLAB_1D_2G.

neutron scalar flux for 1st group energy, ϕ_2^h is the neutron flux for 2nd group energy, and h denotes the size of grid. We assume that $h_1 = 450/18 = 25cm$, $h_2 = h_1/2$, $h_3 = h_1/4$, $h_4 = h_1/8$, $h_5 = h_1/16$, and $h_6 = h_1/32$ are presented as different refined grids, respectively. The point $x_{mid} = 225cm$ in X axis is selected as test point. HARMONY-2 is executed multiple times depending on the different meshes. The calculated results are postprocessed, and the value of the flux on observed point is taken out and normalized, where the *Ratio*1 and *Ratio*2 denote the numerical solution convergence of H^0 and H^1, respectively, as follows:

$$Ratio1 = \frac{\left\|\varphi^{h_i} - \varphi^{h_{i+1}}\right\|_{H^0}}{\left\|\varphi^{h_{i+1}} - \varphi^{h_{i+2}}\right\|_{H^0}},$$
$$Ratio2 = \frac{\left\|\varphi^{h_i} - \varphi^{h_{i+1}}\right\|_{H^1}}{\left\|\varphi^{h_{i+1}} - \varphi^{h_{i+2}}\right\|_{H^1}}. \quad (34)$$

4.2. Test Results and Discussions. The calculation results of the four modes eigenvalues are shown in Table 2 and compared with the results of [14]. Table 2 indicates that the maximum absolute error between the results of our program and lecture [14] is very small. In addition, the neutron flux of ϕ_1, ϕ_2 in 18 equal elements mesh are shown in Figures 5 and 6, respectively, which indicate that the distribution of the neutron scalar flux is geometrically symmetric. The three verification criterions are applied to verification test for the numerical solution of the multigroup neutron diffusion equations. Table 3 shows that the convergence of H^0 on the neutron flux value of x_{mid} point in different meshes. In Table 3, the convergence order approximates to 4. The derivative values of neutron flux on x_{mid} point in different refined grids are listed in Table 4. As the mesh is encrypted, the results are about 2.0 order accuracy in Table 4. The calculation results of Max-eigenvalue in different refined grids are shown in Table 5. Figure 4 shows that Max-eigenvalue converges with the mesh encryption. When the mesh is divided into 72 equal parts, the k_{eff} converges to around 9.86943E-01. Those results indicate that calculation of HARMONY-2 satisfies the three verification criterions constructed by the proposed method in this paper.

Table 4: The derivative values of neutron flux on x_{mid} in different refined grids.

Mesh	Element	The derivative of neutron flux value of x_{mid} for 1th group energy.	*Ratio2*	The derivative of neutron flux value of x_{mid} for 2th group energy.	*Ratio2*
h_1	18	0.80710858E-04		0.8071363955E-04	
h_2	36	0.42252071E-04	0.19102225E+01	0.4224298523E-04	0.19106992E+01
h_3	72	0.21143797E-04	0.19983199E+01	0.2118749068E-04	0.19937701E+01
h_4	144	0.10581871E-04	0.19981151E+01	0.1061129701E-04	0.19966919E+01
h_5	288	0.79545852E-05	0.13302858E+01	0.7929059044E-05	0.13382795E+01

Table 5: The calculation results of Max-eigenvalue in different refined grids.

Mesh	h_1	h_2	h_3	h_4	h_5	h_6
Element	18	36	72	144	288	576
Max-eigenvalue	9.86988E-01	9.86954E-01	9.86945E-01	9.86943E-01	9.86943E-01	9.86942E-01

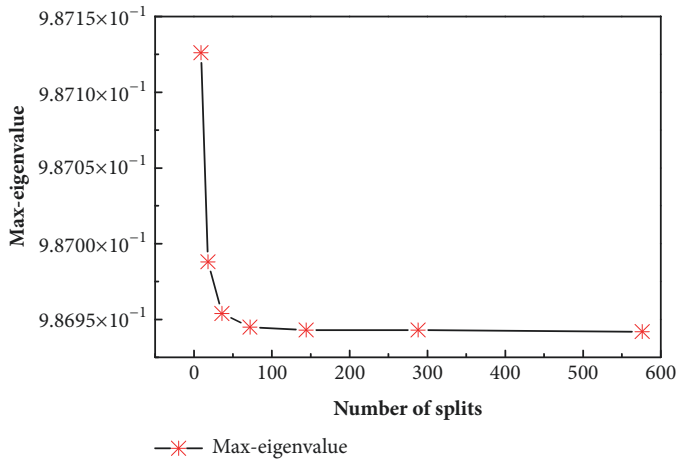

Figure 4: The convergence of the Max-eigenvalue.

Table 2 actually reflects a comparison method, which is commonly used in verification tests. Although this method is easy to perform by comparing with the reference values, it is difficult to get reference values for some newly investigated programs. Our method not only is more rigorous than traditional comparison methods, but also does not need reference solutions, exact solutions, or reference values etc. Though our method is based on convergence properties of numerical solution, it is not difficult to get these convergence properties from the other second-order elliptic partial differential equations by our method. Clearly, our method can be applied to verify the programs of these equations. Therefore, our method is very helpful in alleviating oracle problem and improving the quality of scientific software.

5. Conclusion

In this paper, we proposed a Richardson Extrapolation-based verification method, which used the Richardson Extrapolation on different levels of mesh refinement to estimate the convergences of the numerical solution, and we constructed the verification criterions based on these properties. Compared with traditional methods, our method did not need the exact solution, analysis solutions, and reference solutions as expected values. The calculation program of multigroup neutron diffusion equation was used as an example of verification test. The results show that the numerical convergences accord with the verification criterion. Though the test experiment indicated the validity of our method for alleviating the oracle problem, further research might explore the application of our method to test other programs for solving second-order elliptic partial differential equations.

Data Availability

(1) The data of HARMONY-2 code used to support the findings of this study are available from the corresponding author upon request. (2) The data of Table 1 supporting the calculation of HARMONY-2 in this paper are from previously reported studies and datasets, which have been cited in [14]. The processed data are available from the corresponding author upon request. (3) The data of Tables 2, 3, 4, and 5 are

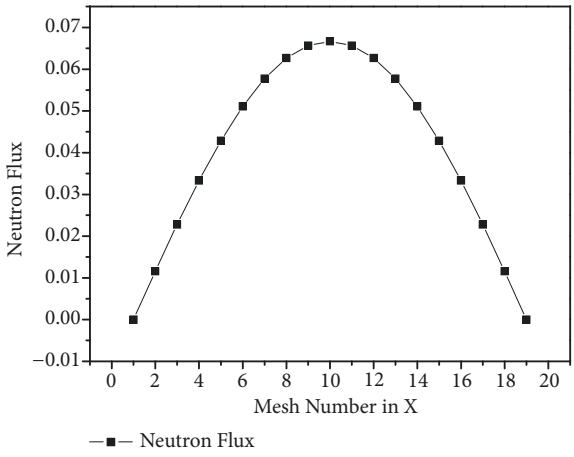

FIGURE 5: ϕ_1 in grid with 18 equal elements.

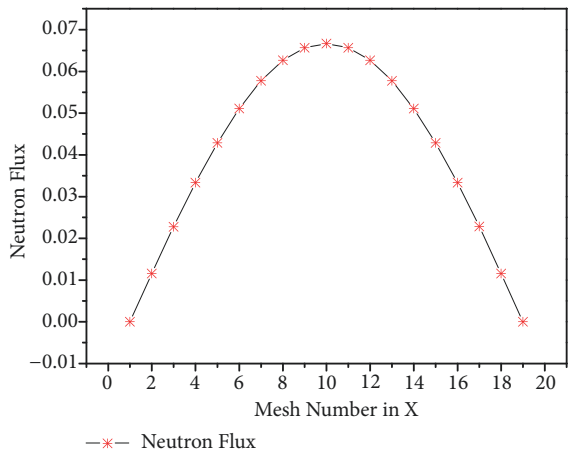

FIGURE 6: ϕ_2 in grid with 18 equal elements.

the calculation results of HARMNOY-2, which are used to support the findings of this study and are available from the corresponding author upon request. (4) The data of Figures 1–6 are the calculation results of HARMNOY-2, which are used to support the findings of this study and are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 11805093. The authors would like to thank Dr. Xie Jinsen of University of South China for providing advice on program of HARMONY-2.

References

[1] W. L. Oberkampf and T. G. Trucano, "Verification and validation benchmarks," *Nuclear Engineering and Design*, vol. 238, no. 3, pp. 716–743, 2008.

[2] A. A. Staf, *AIAA guide for the verification and validation of computational fluid dynamics simulations*, American Institute of Aeronautics & Astronautics, USA, 1998.

[3] E. J. Weyuker, "On testing non-testable programs," *The Computer Journal*, vol. 25, no. 4, pp. 465–470, 1982.

[4] W. E. Howden, *A survey of dynamic analysis methods*, University of Victoria, Department of Mathematics, 1978.

[5] B. Butler, M. Cox, A. Forbes, S. Hannaby, and P. Harris, "A methodology for testing classes of approximation and optimization software," in *Quality of Numerical Software*, IFIP Advances in Information and Communication Technology, pp. 138–151, Springer, Boston, Mass, USA, 1997.

[6] J. Harrison, "Formal proof—theory and practice," *Notices of the AMS*, vol. 55, no. 11, pp. 1395–1406, 2008.

[7] P. J. Roache, "Code verification by the method of manufactured solutions," *Journal of Fluids Engineering*, vol. 124, no. 1, pp. 4–10, 2002.

[8] T. Y. Chen, S. C. Cheung, and S. M. Yiu, "Metamorphic testing: a new approach for generating next test cases," Tech. Rep. HKUST-CS98-01, Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong, 1998.

[9] S. Segura, G. Fraser, A. B. Sanchez, and A. Ruiz-Cortes, "A Survey on Metamorphic Testing," *IEEE Transactions on Software Engineering*, vol. 42, no. 9, pp. 805–824, 2016.

[10] M. Imgrund, *ANSYS® Verification Manual*, Swanson Analysis Systems, Inc, 1992.

[11] P. J. Roache, *Verification and validation in computational science and engineering*, Hermosa, Calif, USA, 1998.

[12] N. Massarotti, M. Ciccolella, G. Cortellessa, and A. Mauro, "New benchmark solutions for transient natural convection in partially porous annuli," *International Journal of Numerical Methods for Heat & Fluid Flow*, vol. 26, no. 3-4, pp. 1187–1225, 2016.

[13] F. Arpino, A. Carotenuto, M. Ciccolella, G. Cortellessa, N. Massarotti, and A. Mauro, "Transient natural convection in partially porous vertical annuli," *International Journal of Heat and Technology*, vol. 34, no. 2, pp. S512–S518, 2016.

[14] K. P. Singh, S. B. Degweker, R. S. Modak, and K. Singh, "Iterative method for obtaining the prompt and delayed alpha-modes of the diffusion equation," *Annals of Nuclear Energy*, vol. 38, no. 9, pp. 1996–2004, 2011.

[15] C. J. Roy, "Review of code and solution verification procedures for computational simulation," *Journal of Computational Physics*, vol. 205, no. 1, pp. 131–156, 2005.

[16] P. J. Roache, "Verification of codes and calculations," *AIAA Journal*, vol. 36, no. 5, pp. 696–702, 1998.

[17] A. V. Avvakumov, P. N. Vabishchevich, A. O. Vasilev, and V. F. Strizhov, "Solution of the 3D neutron diffusion benchmark by FEM," in *Proceedings of the International Conference on Large-Scale Scientific Computing*, vol. 10665 of *Lecture Notes in Computer Science*, pp. 435–442, Springer.

[18] H. Sekimoto, *Nuclear Reactor Theory. Center of Excellence—Innovative Nuclear Energy Systems for Sustainable Development of the World (COE-INES)*, Tokyo Institute of Technology, Tokyo, Japan, 2007, Part 2.

[19] R. Zanette, C. Z. Petersen, M. Schramm, and J. R. Zabadal, "A modified power method for the multilayer multigroup two-dimensional neutron diffusion equation," *Annals of Nuclear Energy*, vol. 111, pp. 136–140, 2018.

[20] C. Ceolin, M. Schramm, Vilhena, and B. E. J. Bodmann, "On an evaluation of the continuous flux and dominant Eigenvalue

problem for the steady state multi-group multi-layer neutron diffusion equation," in *Proceedings of the International Nuclear Atlantic Conference - INAC 2013*, 2013.

[21] S. Han, S. Dulla, and P. Ravetto, "Computational methods for multidimensional neutron diffusion problems," *Science and Technology of Nuclear Installations*, vol. 2009, Article ID 973605, 11 pages, 2009.

[22] A. Hébert, D. Sekki, and R. Chambon, "A User Guide for DONJON Version4," Tech. Rep. IGE-300, École Polytechnique de Montréal Montréal QC, Canada, 2013.

[23] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, ISBN 0-89871-407-9, 1998, http://www.caam.rice.edu/software/ARPACK.