Nuclear Engineering and Technology xxx (xxxx) xxx

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

STRAUM-MATXST: A Code System for Multi-group Neutron-Gamma Coupled Transport Calculation with Unstructured Tetrahedral Meshes

MyeongHyeon Woo, Ser Gi Hong*

Department of Nuclear Engineering, Hanyang University, 222 Wangsimni-ro, Seongdong-gu, Seoul, 04763, South Korea

ARTICLE INFO

Article history:
Received 21 January 2022
Received in revised form
14 June 2022
Accepted 3 July 2022
Available online xxx

Keywords:
STRAUM
Discrete ordinates transport calculation
Krylov subspace method
Block preconditioners for thermal upscattering
MATXST

ABSTRACT

In this paper, a new multi-group neutron-gamma transport calculation code system STRAUM-MATXST for complicated geometrical problems is introduced and its development status including numerical tests is presented. In this code system, the MATXST (MATXS-based Cross Section Processor for S_N Transport) code generates multi-group neutron and gamma cross sections by processing MATXS format libraries generated using NJOY and the STRAUM (S_N Transport for Radiation Analysis with Unstructured Meshes) code performs multi-group neutron-gamma coupled transport calculation using tetrahedral meshes. In particular, this work presents the recent implementation and its test results of the Krylov subspace methods (i.e., Bi-CGSTAB and GMRES(m)) with preconditioners using DSA (Diffusion Synthetic Acceleration) and TSA (Transport Synthetic Acceleration). In addition, the Krylov subspace methods for accelerating the energy-group coupling iteration through thermal up-scatterings are implemented with new multi-group block DSA and TSA preconditioners in STRAUM.

1. Introduction

The accurate solutions of the Boltzmann transport equation for neutral particles (e.g., neutron and photon) are essential for many engineering fields such as radiation shielding and activation analysis, nuclear reactor analysis, and medical applications. Therefore, there have been much efforts to develop the radiation transport calculation codes using unstructured meshes [1,2]. The STRAUM (S_N Transport for Radiation Analysis with Unstructured Meshes) code is also a deterministic code which has been developed by the authors for radiation transport calculation with S_N method and unstructured tetrahedral meshes. The purpose of the STRAUM code development is to provide the detailed forward and adjoint solution results [3] of the neutron and gamma coupled multi-group transport equation for complicated geometrical problems. In STRAUM, Discontinuous Finite Element (DFEM) [4] and LDEM-SCB (Linear Discontinuous Expansion Method with Subcell Balances) [5] spatial discretization methods were employed to solve the transport equation with unstructured tetrahedral meshes. And the DSA method which was derived by discretizing the continuous diffusion equation in a consistent way with LDEM-SCB (1) was successfully developed and applied [6] to accelerate the scattering source iteration. Since the 2000s, the Krylov method has become one of the standard iterative method for effectively solving the discrete ordinates transport equation rather than the conventional scattering source iteration [7]. Recently, we have implemented two Krylov subspace methods (i.e., Bi-CGSTAB and GMRES(m)) to improve the stability and speed of convergence of the within-group scattering iteration in the STRAUM code. In addition, the Krylov space methods have been extended to effectively solve multi-group coupling calculation through thermal up-scatterings [8]. For the multi-group coupling iteration, the diagonal within-group and lower triangular block Gauss-Seidel preconditioners in energy groups have been suggested and implemented based on DSA and TSA for the LDEM-SCB(1) discretization option. Then, the numerical performance of each method was analyzed in detail for both simple and realistic reactor level test problems.

Currently, the STRAUM code uses unstructured tetrahedral meshes generated by Gmsh [9] after importing a CAD file but the Gmsh-generated mesh files are not consistent in the STRAUM code. So, recently, we have written an in-house program to automatically generate the mesh files for STRAUM by processing the meshes files generated by Gmsh.

Originally, the STRAUM code used neutron-gamma coupled cross sections of the ISOTXS files which were generated using the

* Corresponding author. Department of Nuclear Engineering, Hanyang University, 222 Wangsimni-ro, Seongdong-gu, South Korea.
E-mail address: hongsergi@hanyang.ac.kr (S.G. Hong).

https://doi.org/10.1016/j.net.2022.07.002

TRANSX code. However, the ISOTXS format has some limitations in that the syntax is not extendable, so an only limited number of data are included, and the background cross section is not included. So, recently, we have developed the MATXST code [10] (MATXS-based Cross Section Processor for S_N Transport) for generating multi-group cross sections for STRAUM. The MATXST code reads MATXS-formatted cross section files and generates a multi-group cross section for the STRAUM code using transport corrections and the Bondarenko iteration for self-shielding effect. The coupled calculation of STRAUM and MATXST was applied to a simplified reactor shielding problem and the results were compared to the MCNP calculation results for numerical validation.

2. Theory and methodology

2.1. Discrete ordinate method

The steady-state neutral particle transport equation is given by

$$\widehat{\Omega}\cdot\nabla\psi(\overrightarrow{r},E,\widehat{\Omega})+\Sigma_t(\overrightarrow{r},E)\psi=q(\overrightarrow{r},E,\widehat{\Omega}), \tag{1}$$

where ψ is angular flux, Σ_t is total macroscopic cross section, and q is source term. In a fixed source problem, the source term is defined as the sum of scattering and external source terms as follows:

$$q=q_s+q_{ex}, \tag{2}$$

where q_{ex} represents the external source term, and q_s represents the scattering source term which is given by

$$q_s=\int_0^{\infty}dE'\int_{4\pi}d\widehat{\Omega}'\Sigma_s(\overrightarrow{r},E'\rightarrow E,\ \widehat{\Omega}'\cdot\widehat{\Omega})\psi(\overrightarrow{r},E',\widehat{\Omega}'). \tag{3}$$

Applying the multi-group and the discrete ordinates approximations to the transport equation (i.e., Eq. (1)) with Legendre expansion to the scattering anisotropy gives the following multi-group discrete ordinates transport equation:

$$\begin{aligned}&\widehat{\Omega}_n\cdot\nabla\psi_n^g(\overrightarrow{r})+\Sigma_t^g(\overrightarrow{r})\psi_n^g(\overrightarrow{r})\\&=\sum_{g'=1}^{G}\sum_{\ell=0}^{L}\sum_{m=-\ell}^{\ell}(2\ell+1)\Sigma_{s,\ell}^{g'\rightarrow g}R_{\ell m}(\widehat{\Omega})\phi_{\ell m}(\overrightarrow{r})+q_{ex,n}^g(\overrightarrow{r}),\end{aligned} \tag{4}$$

where

$\psi_n^g\equiv\psi^g(\widehat{\Omega}_n)$,

Σ_t^g = macroscopic total cross section for group g,
$\Sigma_{s,\ell}^{g'\rightarrow g}$ = ℓ th moment of the scattering cross section from g′ to g,
$R_{\ell m}$ = tesseral (or real) spherical harmonics function,
$\phi_{\ell m}$ = flux moment corresponding to $R_{\ell m}$.
In the S_N method, the flux moments are calculated by numerical integration using the quadrature rules over the angular space of unit sphere as follows:

$$\phi_{\ell m}=\int_{4\pi}R_{\ell m}(\widehat{\Omega})\psi(\widehat{\Omega})d\Omega\cong\sum_{n=1}^{N}w_n\psi_nR_{\ell m}(\widehat{\Omega}_n), \tag{5}$$

where $\widehat{\Omega}_n$ is a discrete ordinate direction, w_n is the corresponding weight associated with this ordinate direction, and N is the number of the discrete ordinate directions. In the STRAUM code, the Gauss-Chebyshev quadrature is used with a weight normalization to 1.0 as follows:

$$\sum_{n=1}^{N}w_n=1.0. \tag{6}$$

The transport equation expressed in Eq. (4) can be written by an operator form as

$$\boldsymbol{L}\psi=\boldsymbol{MS}\phi+q_{ex}=Q, \tag{7}$$

where $\boldsymbol{L}$ is the transport operator comprised of streaming and total collision terms, $\boldsymbol{M}$ is the moment-to-discrete operator which maps the moments into the discrete direction dependent quantities, and $\boldsymbol{S}$ is the scattering operator which calculates the scattering source with the moments. Explicitly defining the operators as matrix and solving the transport equation in a matrix-vector form requires a lot of computer memory and computation time, and so the matrix-free manner is generally applied in transport theory. In particular, the action of $\boldsymbol{L}^{-1}$ is implicitly defined as a transport sweeping operation that updates the outgoing fluxes using the incoming fluxes while crossing all the cells ordered in the directions. As shown in Eq. (7), the scattering operator requires the moments which are obtained using the discrete-to-moment operator defined as follows:

$$\phi=\boldsymbol{D}\psi. \tag{8}$$

The sweep operation is performed based on the sorted order for a specific direction. As shown in Fig. 1, the sweeping order in a specific direction can be expressed as a directed acyclic graph (DAG), and the sweeping order is derived through a topological sorting for this graph. To parallelize sweep operation with dependency between tasks, we implemented the wavefront parallelism based on thread pool using the Taskflow library [11].

2.2. Implementation of the Krylov Subspace Methods

In order to apply the Krylov subspace method, it is necessary to rearrange Eq. (7) into a suitable form of $\boldsymbol{A}x=b$. For this purpose, the operator $\boldsymbol{L}^{-1}$ is applied to Eq. (7) and taking the discrete-to-moment operator can leads to the following equation.

$$\left(\boldsymbol{I}-\boldsymbol{DL}^{-1}\boldsymbol{MS}\right)\phi=\boldsymbol{DL}^{-1}Q. \tag{9}$$

By introducing $\boldsymbol{T}=\boldsymbol{DL}^{-1}$ operator, Eq. (9) can be expresses as

$$(\boldsymbol{I}-\boldsymbol{TMS})\phi=\boldsymbol{T}Q. \tag{10}$$

This equation can be written in the following simple form:

$$\boldsymbol{A}\phi=b, \tag{11}$$

where

$$\begin{aligned}&\boldsymbol{A}=(\boldsymbol{I}-\boldsymbol{TMS}),\\&b=\boldsymbol{T}Q.\end{aligned} \tag{12}$$

Two Krylov subspace methods, i.e., Restarted GMRES(m) [12] and BiCGSTAB [13] were implemented in the STRAUM code. As mentioned above, the Krylov subspace methods were implemented in a matrix-free manner using the operator form. The flux moments are defined as one explicit vector and stored in a one-dimensional array. The convergence and robustness of the Krylov subspace methods for the linear system can be improved with a preconditioner by reducing the spectral radius. By using the a preconditioner $\boldsymbol{W}$, the left-preconditioned system for Eq. (11) can be written as

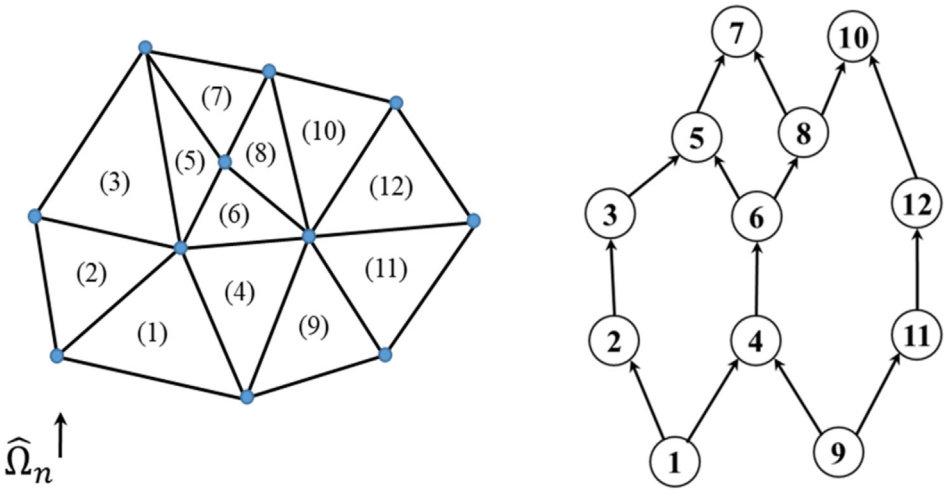

Fig. 1. Illustration for spatial meshes and a directed acyclic graph aligned for a specific direction.

$$\boldsymbol{WA}\phi = \boldsymbol{W}b. \tag{13}$$

The Richardson iteration for the linear system given by Eq. (11) is expressed by

$$\phi^{(n+1/2)} = (\boldsymbol{I} - \boldsymbol{A})\phi^{(n)} + b. \tag{14}$$

Actually, the Richardson iteration (i.e., Eq. (14)) represents the standard source iteration. The subtraction of Eq. (14) from Eq. (11) after multiplication of $\boldsymbol{A}$ to Eq. (14) gives the following error equation:

$$\boldsymbol{A}\varepsilon^{(n+1/2)} = (\boldsymbol{I} - \boldsymbol{A})\left(\phi^{(n+1/2)} - \phi^{(n)}\right), \tag{15}$$

where the error is defined by $\varepsilon^{(n+1/2)} = \phi - \phi^{(n+1/2)}$. With the substitution of Eq. (12) into Eq. (15), the error equation for the S_N equation can be written as

$$(\boldsymbol{I} - \boldsymbol{TMS})\varepsilon^{(n+1/2)} = \boldsymbol{TMS}\left(\phi^{(n+1/2)} - \phi^{(n)}\right). \tag{16}$$

As can be seen from Eq. (16), the error equation has the same complexity as solving the original equation, so the error vector $\varepsilon^{(n+1/2)}$ is approximately obtained in the synthetic acceleration methods as follows:

$$\varepsilon^{(n+1/2)} \approx \tilde{\boldsymbol{W}}\left(\phi^{(n+1/2)} - \phi^{(n)}\right), \tag{17}$$

where $\tilde{\boldsymbol{W}}$ is an approximate operator of $(\boldsymbol{I} - \boldsymbol{TMS})^{-1}\boldsymbol{TMS}$. Then, the next iterate flux moments are updated for the Richardson iteration (i.e., Eq. (14)) as follows:

$$\phi^{(n+1)} = \phi^{(n+1/2)} + \varepsilon^{(n+1/2)}. \tag{18}$$

A simple algebra can show that this synthetic acceleration can be equivalent to the preconditioned Richardson iteration and the preconditioner $\boldsymbol{W}$ in Eq. (13) is equal to $\boldsymbol{I} + \tilde{\boldsymbol{W}}$. The STRAUM code has two preconditioners, Diffusion Synthetic Acceleration (DSA) and β-Transport Synthetic Acceleration (β-TSA) for the within-group iteration.

2.2.1. Diffusion Synthetic Acceleration (DSA)

The DSA preconditioner approximates the error equation by using the diffusion equation as follows:

$$\varepsilon^{(n+1/2)} \approx \boldsymbol{PC}^{-1}\boldsymbol{RS}\left(\phi^{(n+1/2)} - \phi^{(n)}\right), \tag{19}$$

and

$$\boldsymbol{C} \equiv -\nabla \cdot D\nabla + \Sigma_t - \Sigma_s, \tag{20}$$

where the $\boldsymbol{C}$ operator represents the diffusion operator. In Eq. (20), $\boldsymbol{R}$ is the restriction operator which extracts the isotropic component from the scattering source, and $\boldsymbol{P}$ is the prolongation operator which prolongates the scalar flux component of the error onto the full moments. In the STRAUM code, the continuous diffusion equation is discretized in a consistent way with the LDEM-SCB(1) discretization of the transport equation [6], which gives a linear discontinuous discretization. Originally, the discretized diffusion equation was solved using Gauss-Seidel like iteration coupled with the FMR (Fine Mesh Rebalance) method [14]. On the other hand, the current STRAUM code explicitly sets up the whole matrix with consideration of its sparsity for the discretized diffusion equation and solves it using the Krylov subspace methods with the Eigen library [15].

2.2.2. β-Transport Synthetic Acceleration (β-TSA)

The β-TSA [16] calculates the error vector by solving an approximate transport equation in which the within group scattering term is modified by a fraction of the parameter β and the total cross section is also reduced by a β fraction of scattering cross section as follows:

$$\varepsilon^{(n+1/2)} \approx \left(\boldsymbol{I} - \boldsymbol{DL}_{\beta,TSA}^{-1}\boldsymbol{MS}_{\beta,lo}\right)^{-1}\boldsymbol{TMS}\left(\phi^{(n+1/2)} - \phi^{(n)}\right), \tag{21}$$

where β-TSA transport operator $\boldsymbol{L}_{\beta,TSA}$ and low-order scattering operator $\boldsymbol{S}_{\beta,lo}$ are defined as

$$\boldsymbol{L}_{\beta,TSA}(\cdot) \equiv (\hat{\Omega}\cdot\nabla + (\Sigma_t - \beta\Sigma_s))(\cdot), \tag{22}$$

and

$$\boldsymbol{S}_{\beta,lo} \equiv (1-\beta)\boldsymbol{SR}_{lo}. \tag{23}$$

In Eq. (23), the restriction operator $\boldsymbol{R}_{lo}$ extracts the lowest order moment. In general, TSA generally considers a low order (lo) quadrature set and low order anisotropic scattering than the

original transport equation. The STRAUM code considers the 2x2 Gauss-Legendre quadrature set and only P_0 scattering, and solves the TSA with the BiCGSTAB method in a matrix-free way.

2.3. Multi-group Krylov methods with energy block preconditioners

In the multi-group problem, the Gauss-Seidel iteration method is commonly used to solve the energy group coupling structure, which can be represented with the iteration index k as follows:

$$(\boldsymbol{I}-\boldsymbol{T}_g\boldsymbol{M}\boldsymbol{S}_{gg})\phi_g^{(k+1)}=\boldsymbol{T}_g\boldsymbol{M}\left(\sum_{g'=1}^{g-1}\boldsymbol{S}_{gg'}\phi_{g'}^{(k+1)}+\sum_{g'=g+1}^{G}\boldsymbol{S}_{gg'}\phi_{g'}^{(k)}+q_g^{ex}\right), \tag{24}$$

where $\boldsymbol{S}_{gg'}$ represents the scattering operator from group g′ to g.

As shown in Eq. (24), the GS method solves sequentially from high energy to low energy through energy group sweeping, but if there exists up-scattering, the extra-iterations additional to the single energy group sweeping are required for full convergence. Since up-scattering iteration can take a lot of time, the multi-group Krylov method can be applied to iteratively solve the energy groups coupled through up-scatterings [8]. For this purpose, the multi-group transport equations can be written in the following single matrix-vector form for all the energy groups in an energy group chunk *GC* comprised of several energy groups:

$$(\boldsymbol{I}-\boldsymbol{T}_{GC}\boldsymbol{M}_{GC}\boldsymbol{S}_{GC})\phi_{GC}^{(k+1)}=\boldsymbol{T}_{GC}\boldsymbol{M}_{GC}\left(\mathbf{S}_{incom}\phi_{incom,GC}^{(k+1)}+q^{ex}\right). \tag{25}$$

In Eq. (25), "*income*" refers to the energy groups that are not included in the group chunk *GC* and give the scattering contribution to the group chunk *GC*. In Eq. (25), the multi-group operator $\boldsymbol{T}_{GC}$ is related to the group-wise sweeping where a single angular sweeping is performed for each group. In STRAUM, the upper groups having only down-scattering are sequentially calculated with the conventional GS iteration while the other groups with upscattering is treated as one chunk. For example, Fig. 2 illustrates the case where each group in the down scattering range is considered as a group chunk and the last three groups in the upscattering range are considered a single group chunk. STRAUM uses BiCGSTAB and GMRES(m) as the Krylov subspace methods to solve Eq. (26).

In this work, two preconditioners for the multi-group calculation in a group chunk are suggested and implemented to accelerate the Krylov subspace methods. To explain the multi-group preconditioners, it is convenient to represent them using the block operators of energy group in a group chunk as follows:

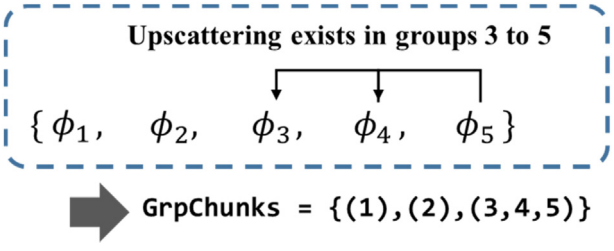

Fig. 2. Illustration of group chunks for multi-group Krylov method.

$$\boldsymbol{W}_{MG}=\begin{pmatrix}\boldsymbol{W}_{11} & \boldsymbol{W}_{22} & \cdot & \boldsymbol{W}_{1C}\\ \boldsymbol{W}_{21} & \boldsymbol{W}_{22} & \cdot & \boldsymbol{W}_{2C}\\ \cdot & \cdot & \cdot & \cdot\\ \boldsymbol{W}_{C1} & \boldsymbol{W}_{C2} & \cdot & \boldsymbol{W}_{CC}\end{pmatrix}, \tag{26}$$

where C represents the number of energy groups in a group chunk. As shown in Eq. (26), the multi-group preconditioner can also include the upscattering elements, but considering the upscattering is expected to be inefficient even in the diffusion calculations. Therefore, this work suggested two preconditioners that consider only the within group or the down scattering term without upscattering and implemented them in STRAUM. These multi-group preconditioners can be expressed as Eqs. (27) and (28), respectively, as follows:

a) Diagonal within-group block DSA (DiagDSA) and TSA (DiagTSA) preconditioners in energy.

$$\boldsymbol{W}_{gg'}^{Diag}=\begin{cases}\delta_{gg'}\boldsymbol{C}_g & \text{for } DSA\\ \delta_{gg'}\left(\boldsymbol{I}-\boldsymbol{DL}_{g,\beta,TSA}^{-1}\boldsymbol{M}\boldsymbol{S}_{g,\beta,lo}\right) & \text{for } TSA\end{cases}, \tag{27}$$

where

$$\boldsymbol{C}_g=-\nabla\cdot D\nabla+\Sigma_{t,g}-\Sigma_{s,gg},$$
$$\boldsymbol{L}_{g,\beta,TSA}=\left(\widehat{\Omega}\cdot\nabla+\left(\Sigma_{t,g}-\beta\Sigma_{s,gg}\right)\right),$$

and

$$\boldsymbol{S}_{gg,\beta,lo}=(1-\beta)\boldsymbol{S}_{gg}\boldsymbol{R}_{lo}.$$

b) Lower triangular block DSA (LtriDSA) and TSA (LtriTSA) preconditioners in energy.

$$\boldsymbol{W}_{gg'}^{Ltri}=\begin{cases}0 & \text{for } g<g'\\ \boldsymbol{W}_{gg}^{Diag} & \text{for } g=g'\\ -\boldsymbol{S}_{gg'} & \text{otherwise}\end{cases}. \tag{28}$$

3. MATXS file processing code development

The MATXST code which was written using C++ to provide neutron-gamma coupled multi-group cross sections for STRAUM calculates a self-shielded cross section in an infinite medium by performing the Bondarenko iteration on the data contained in the MATXS file. After reading MATXS files generated using the NJOY code [17], the MATXST code performs a balance check between the P_0 total cross section and the sum of the sub-reactions for reference data corresponding to the infinitely diluted case at the reference temperature. Next, the MATXST code performs the Bondarenko iteration to calculate σ_0 for each materials, and constructs transport table for discrete ordinates transport calculation.

The P_0 total cross section is explicitly given in MATXS file, but it is reconstructed by summing of partial reactions. In the case of the reference data given at a reference temperature and infinitely dilution, there is no difference between the sum of the sub-reactions and the P_0 total cross section explicitly stored in the

Table 1
Configuration list for neutron P_0 total cross section.

Symbol	MT number	Description	Reaction Identifier
$\sigma_{t,o}$	1	P_0 total	ntot0
σ_{el}	2	Elastic scattering	nelas
σ_{in}	4	Inelastic scattering (Sum of the MT = 51–91)	ninel (Sum of n01, n02, n03, …, n40, ncn)
σ_{ftot}	18	Total fission (Sum of MT 19,20,21 and 38)	nftot
σ_{capt}	102–109 111–117	Sum of capture reactions	ng, np, nd, nt, nh, na, n2a, n3a, n2p, npa, nt2a, nd2a, npd, npt, nda
σ_{misc}	5	Sum of all reactions not given explicitly in another MT number	nx
	16	(n,2n)	n2n
	17	(n,3n)	n3n
	37	(n,4n)	n4n
	11, 22–25, 28–30, 32–36, 41,42,44,45	Production cross section of neutrons + various particles	n2nd, nna, nn3a, n2na, n3na, nnp, nn2a, n2n2a, nnd, nnt, nnhe3, nnd2a, nnt2a, n2np, n3np, nn2p, nnpa

MATXS file. However, in the other cases, a slight discrepancy can occur between the P_0 total cross section and sum of sub-reactions because the other cases contain differences from the reference only in the following reactions: “nelas”, “n01”, “ng”, and “nftot”. Although these discrepancies do not much affect the transport calculation results, the MATXST code calculates the P_0 total cross section using the following equation:

$$\sigma_{t,o} = \sigma_{el} + \sigma_{in} + \sigma_{ftot} + \sigma_{capt} + \sigma_{misc}, \tag{29}$$

where $\sigma_{t,o}$ is the P_0 total cross section, σ_{el} is the elastic scattering cross section, σ_{in} is the inelastic scattering cross section, σ_{ftot} is the total fission cross section, σ_{capt} is the sum over all kinds of capture cross sections, and σ_{misc} is the sum of the reactions that constitute $\sigma_{t,o}$ but do not belong to the previous classification.

The types and classifiers of the reactions belonging to the P_0 total cross section are summarized in Table 1.

The MATXS file does not contain the total scattering transfer matrix, but instead contains the sub-reaction scattering transfer matrices. Therefore, the total scattering transfer matrix for the ℓ th anisotropy is calculated by considering all the given sub-reaction scattering matrices as follows:

$$\sigma_{s,\ell}^{g'\to g} = \sigma_{el,\ell}^{g'\to g} + \sum_{x\in misc} \sigma_{x,\ell}^{g'\to g} + \sum_{i} \sigma_{inel,i,\ell}^{g'\to g} + \sigma_{inel,c,\ell}^{g'\to g}\ , \tag{30}$$

where

$\sigma_{el,\ell}^{g'\to g}$: elastic scattering transfer matrix,
$\sigma_{x,\ell}^{g'\to g}$: scattering transfer matrix for miscellaneous reaction,
$\sigma_{inel,i,\ell}^{g'\to g}$: inelastic scattering transfer matrix of i th level,
$\sigma_{inel,c,\ell}^{g'\to g}$: inelastic scattering transfer matrix of continuous level.

The MATXST code reconstructs the total cross section by adding the thermal cross section instead of the elastic cross section in the thermal energy region if the user wants to treat thermal data. The transport correction is performed to produce multi-group cross sections for S_N transport calculations. The MATXST code provides

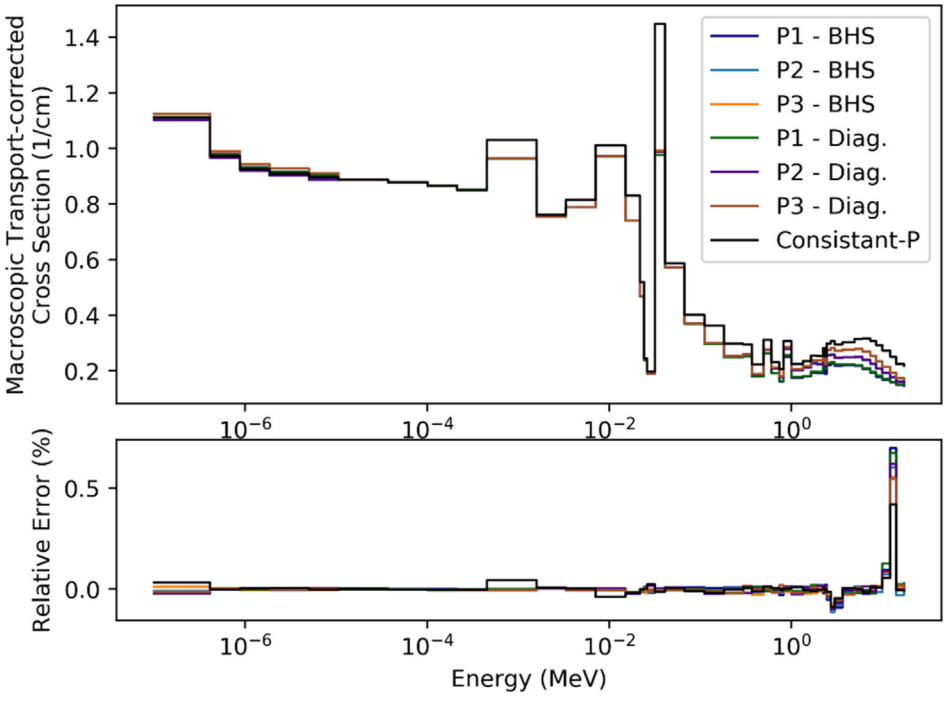

Fig. 3. Transport-corrected P_0 cross section of SS304.

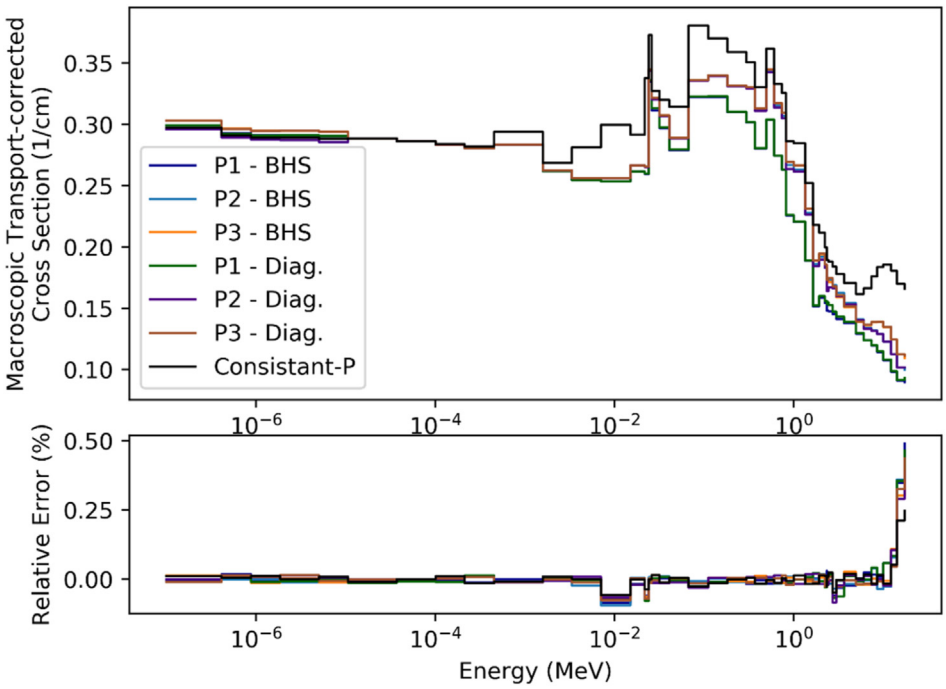

Fig. 4. Transport-corrected P_0 cross section of ZIRLO

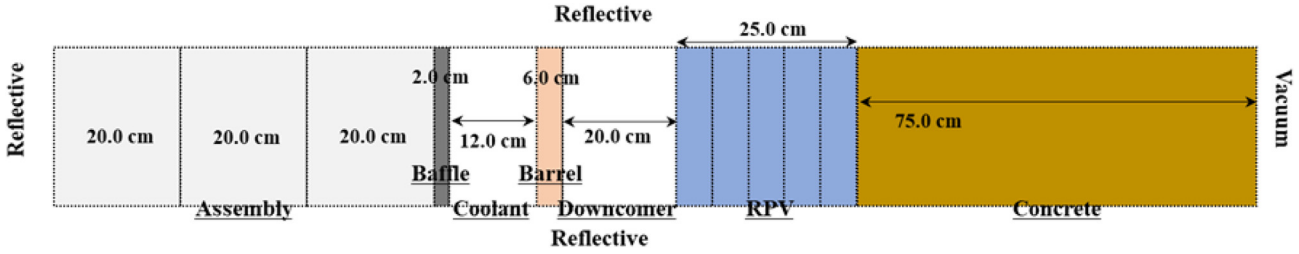

Fig. 5. Layout of verification problem for neutron-gamma coupled cross section.

for the transport correction options in the same way used in the TRANSX2 code (i.e., consistent, inconsistent, diagonal, and Bell-Hansen-Sandmeier (BHS) approximations).

Since photon has no charge, it can be treated numerically in the same way as the neutron in the discrete ordinates method. If a coupled table of neutron and photon is formed by placing the neutron-to-gamma production matrix and the gamma scattering matrix after the neutron data, then the angular flux for gamma produced by neutrons can be evaluated through a single transport calculation. The MATXS files do not provide the total neutron-to-gamma production matrix, but provides each production matrix for several neutron reactions. The MATXS file can store neutron-to-gamma matrix as a function of the background cross section, but the NJOY code provides only the infinitely diluted case. Therefore, the MATXST code considers the self-shielding effect of the gamma production matrix by using the self-shielding factor defined as follows:

$$f_{x,g}(T,\sigma_0) = \frac{\sigma_{x,g}(T,\sigma_0)}{\sigma_{x,g}\left(T_{ref},\infty\right)}, \tag{31}$$

where

$f_{x,g}$: self-shielding factor for reaction x and background cross section σ_0,
$\sigma_{x,g}(T,\sigma_0)$: self-shielded reaction x cross section for background cross section σ_0 and temperature T,
$\sigma_{x,g}(T_{ref},\infty)$: infinitely diluted cross section for reaction x and reference temperature.

Then, the total neutron-to-gamma production matrix is calculated by

$$\sigma_{ng}^{g'\to g} = \sum_{x\in\gamma} f_{x,g'}(T,\sigma_0)\sigma_x^{g'\to g} \quad \text{for all } g' \text{ and } g. \tag{32}$$

In this equation, x represents a specific reaction belonging to the neutron reaction set γ producing gamma.

4. Numerical analysis and results

4.1. MATXST cross section validation problem

In this section, the MATXST code was verified by comparing two multi-group cross section sets generated with the TRANSX2 [18] and MATXST codes and validated by comparing the neutron and gamma fluxes obtained by the STRAUM code with multi-group

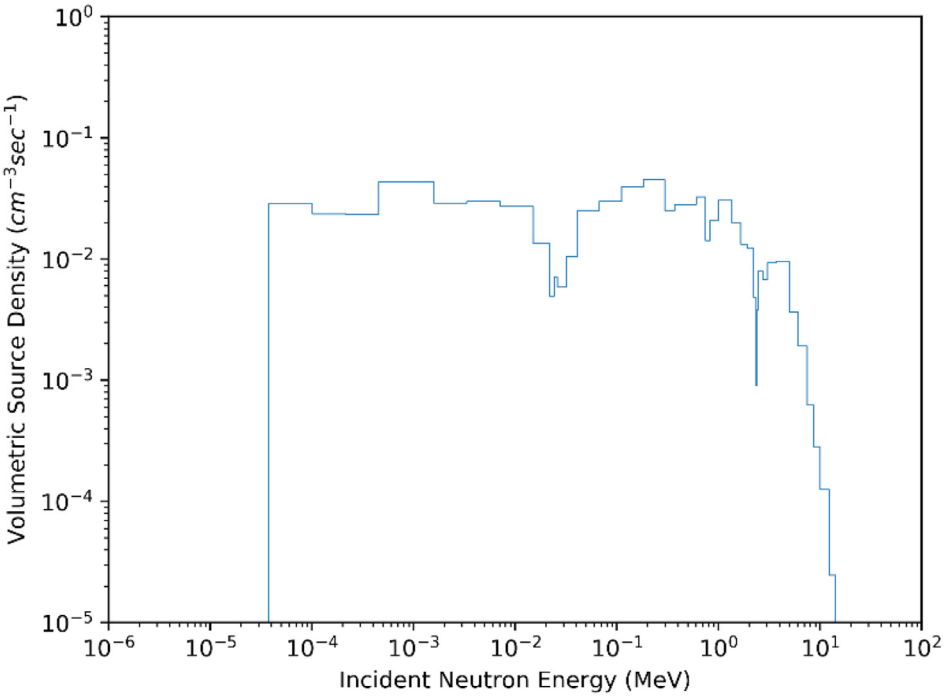

Fig. 6. Neutron source spectrum used for the verification problem for neutron-gamma coupled cross section.

Table 2
Calculation conditions for verification problem with neutron-gamma coupled cross section.

Calculation code	MCNP6.2[a]	STRAUM	
Cross section library	Continuous energy (ACE format)	47n20g BUGLE-96 structure (by MATXST)	47n20g BUGLE-96 structure (by TRANSX)
Anisotropy order of scattering	n/a	P_3	P_3
Gauss-Chebyshev quadrature set (# of azimuthal × polar direction per octant)	n/a	2 × 2	2 × 2

[a] Number of histories for MCNP fixed source calculation is 500,000,000.

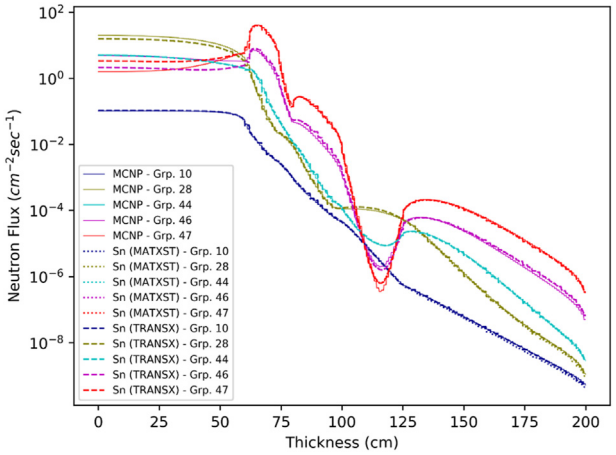

Fig. 7. Neutron group flux distributions for the 1D reactor shielding problem.

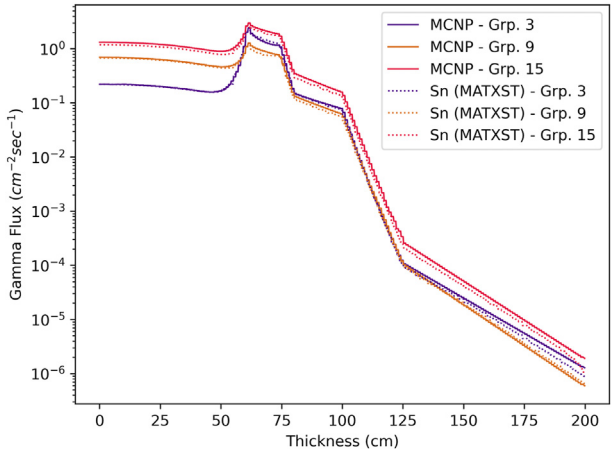

Fig. 8. Gamma group flux distributions for the 1D reactor shielding problem.

cross sections generated with MATXST and by the MCNP6 [19] code for a benchmark problem. The MATXS libraries used for verification of MATXST were generated using ENDF/B-VIII.0 library [20] and NJOY21. The group structures of the library were applied with 27 neutron and 19 gamma groups (27n19g) [21] and the BUGLE-96 [22] structure, which are widely used in shielding calculations. All the nuclides in the library were treated at 300, 500, 700, and 1100 K temperatures, and the background cross sections were set to have 10 grid points (1×10^{10}, 5×10^{8}, 5×10^{7}, 5×10^{6}, 5×10^{5}, 5×10^{4}, 5×10^{3}, 5×10^{2}, 50, 5 b). For the neutron flux weighting, a typical spectrum of thermal + 1/E + fission + fusion (IWT = 10) in NJOY was used while for the photon, the 1/E + rolloffs weighting

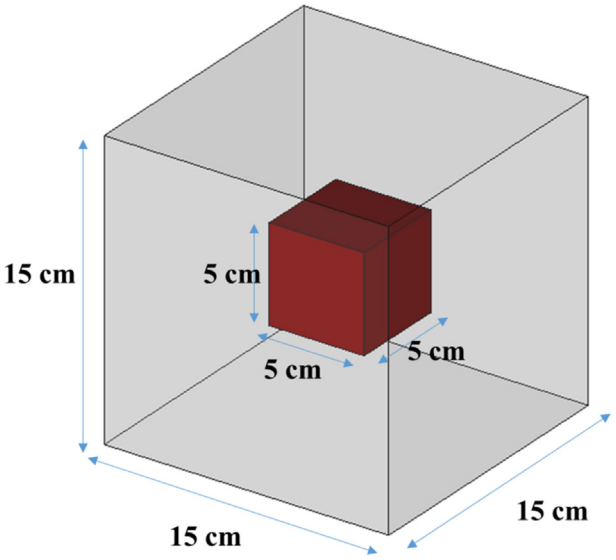

Fig. 9. Simple box problem layout.

function was applied using the (IWT = 3) option. Thermal scattering data were generated only for free gas scattering with cutoff energy of 5.0 eV. The TRANSX2 code was used to process the MATXS files to create ISOTXS files with consideration of transport correction and the Bondarenko self-shielding treatment, and they were finally used for transport calculation using the STRAUM code. Independently, the MATXST code generated a transport table in the STRAUM code's own cross section format by processing the same MATXS files.

In order to verify the MATXST code, the cross sections created with the MATXST and TRANSX2 codes were inter-compared. For this purpose, the macroscopic cross sections of SS304 and ZIRLO with BUGLE-96 group structure were generated using the MATXS files with the TRANSX2 and MATXST codes. The transport-corrected P_0 total cross sections evaluated by the MATXST code for SS304 and ZIRLO are shown in Figs. 3 and 4, respectively. These figures also show the relative discrepancies (%) of the cross sections generated using MATXST and TRANSX2 for each case. The anisotropic order of scattering was considered up to P_3. These figures show the results of the Diagonal and BHS methods converged to the same value as the anisotropic order increases. In the cases of the BHS and Diagonal transport corrections, the macroscopic cross sections of SS304 and ZIRLO generated by MATXST were shown to have a maximum relative discrepancy of 0.7 % and 0.5 %, respectively, in comparison with those obtained using TRANSX2. The tendency of these relative discrepancies is quite similar to that of Consistent-P method that does not modify the total cross section.

Next, a comparison with MCNP6 was performed to verify the MATXST and STRAUM codes. For this purpose, a 1D reactor shielding problem was considered in order to minimize the effects of the mesh size, quadrature set, and anisotropic order of scattering

Table 3
Elapsed time (seconds) and number of sweep operations without preconditioner.

Quadrature Set	Group chunk	$\vec{b}$ setup	Iteration method (# of Sweeps)		
			Richardson	GMRES(30)	BiCGSTAB
2x2	1 = {1}	0.22	1.5 (7)	1.4 (6)	0.9 (4)
	2 = {2}	0.16	3.4 (20)	1.5 (9)	1.4 (8)
	3 = {3}	0.16	2.9 (17)	1.5 (9)	1.1 (6)
	4 = {4,5,6,7}	0.38	108.8 (304)	15.1 (42)	12.1 (32)
3x3	1	0.40	2.8 (7)	2.8 (6)	1.8 (4)
	2	0.37	7.3 (20)	4.0 (9)	3.2 (8)
	3	0.47	6.3 (17)	4.2 (9)	2.5 (6)
	4	0.75	231.8 (304)	34.5 (42)	25.0 (30)
4x4	1	0.64	4.9 (7)	3.9 (5)	3.0 (4)
	2	0.63	14.7 (20)	6.5 (9)	5.3 (8)
	3	0.65	13.3 (17)	6.2 (9)	4.2 (6)
	4	1.50	422.7 (304)	65.6 (42)	44.3 (30)
5x5	1	0.94	7.7 (7)	5.5 (5)	4.6 (4)
	2	0.92	19.5 (20)	9.5 (9)	8.3 (8)
	3	0.99	16.6 (17)	9.4 (9)	6.5 (6)
	4	1.99	620.2 (304)	90.0 (42)	62.0 (30)

Table 4
Elapsed time for 4x4 quadrature set with DSA preconditioner in seconds (DiagDSA preconditioner for the last group chunk).

	Group chunk	DSA-Setup[a)]	Number of Iterations	SweepOperations	DSAPrecond.	Etc.[c)]
Richardson.	1 = {1}	0.049	5	3.6	0.9	0.012
	2 = {2}	0.050	7	4.8	0.7	0.024
	3 = {3}	0.049	6	4.3	0.6	0.020
	4 = {4,5,6,7}	0.189	53	74.6	12.1	0.365
GMRES(30).	1	0.057	5	4.4	0.9	0.021
	2	0.051	6	4.3	0.6	0.030
	3	0.048	6	4.5	0.6	0.029
	4	0.194	20	29.0	5.0	0.261
BiCGSTAB.	1	0.050	2 (4[b)])	3.0	0.7	0.008
	2	0.044	2 (4)	3.0	0.4	0.011
	3	0.045	2 (4)	3.0	0.3	0.010
	4	0.193	7 (14)	20.2	3.2	0.069

[a)] DSA-Setup is elapsed time for constructing explicit DSA matrix.
[b)] Number of sweep operations.
[c)] Etc. is composed of memory copy and simple linear algebra operations.

Table 5
Elapsed time with the DiagTSA in seconds (4x4 and 2x2 quadrature sets for high-order transport and TSA calculations, respectively).

	Group chunk	$\beta = 0.0$	$\beta = 0.3$	$\beta = 0.5$	$\beta = 1.0$
Richardson	1	5.3 ([a])53.8%)	5.0 (52.1%)	4.7 (50.0%)	3.7 (13.0%)
	2	8.2 (63.2%)	10.0 (59.3%)	10.8 (61.4%)	8.1 (15.3%)
	3	5.9 (60.0%)	7.5 (59.7%)	9.2 (61.2%)	7.2 (15.0%)
	4	476.2 (84.5%)	473.5 (83.3%)	477.0 (83.2%)	363.2 (24.8%)
GMRES(30)	1	7.0 (53.7%)	6.8 (55.5%)	6.4 (51.2%)	3.5 (13.5%)
	2	10.8 (64.6%)	9.6 (61.4%)	9.7 (60.9%)	5.6 (14.7%)
	3	7.7 (60.8%)	7.8 (60.4%)	9.9 (61.8%)	4.9 (14.2%)
	4	169.7 (82.6%)	167.0 (82.2%)	155.5 (79.2%)	52.9 (26.0%)
BiCGSTAB	1	3.8 (51.9%)	3.4 (48.1%)	3.9 (42.1%)	2.0 (12.1%)
	2	3.9 (54.3%)	8.0 (61.2%)	7.4 (57.4%)	3.9 (13.2%)
	3	3.9 (54.6%)	4.3 (52.2%)	7.6 (58.5%)	3.6 (13.5%)
	4	117.4 (81.4%)	99.7 (78.8%)	111.8 (76.9%)	40.5 (25.6%)

[a] Fraction of TSA calculation time in the total iteration time.

for S_N transport calculation. As shown in Fig. 5, in this problem, the reactor core represented by three fuel assemblies is sequentially surrounded by 2.0 cm thick baffle, 12 cm thick coolant, 6.0 cm thick barrel, 20.0 cm thick down comer, 25.0 cm reactor pressure vessel (RPV), and 75.0 cm concrete.

In order to generate the neutron source term leading to the consistent neutron leakage to the reactor pressure vessel compared to the eigenvalue calculation, the neutron source was defined by using the spectrum of the baffle region obtained from the eigenvalue mode calculation using MCNP6. The neutron spectrum was obtained to have the BUGLE-96 group structure. The comparative calculations using STRAUM and MCNP6 were performed with fixed source calculations which replace the fuel and moderator by ZIRLO using the same source term both in MCNP and STRAUM. The neutron source spectrum is given in Fig. 6.

In this verification problem, the fuel and coolant were removed from the assembly region and the remaining space was filled with ZIRLO. And the extracted neutron source from the eigenvalue calculation mode using MCNP6 was uniformly distributed in this region. The materials of baffle, barrel, and RPV were set to SS304, and the coolant was assumed to be pure water. For S_N transport calculation, the multi-group cross sections of the BUGLE-96 energy group structure (47 neutron and 20 gamma groups) were generated using the MATXST code with the BHS transport correction. To obtain the reference flux, MCNP fixed source calculation was performed using the same extracted neutron source but with pointwise cross sections. The options used in the calculation are summarized in Table 2.

The scalar flux distributions obtained with STRAUM using TRANSX2 and MATXST-generated multi-group cross sections were compared with those obtained with MCNP6. As shown in Figs. 7 and 8 it can be seen that the results of MCNP and the STRAUM code using the MATXST cross sections show good agreements over the entire region. From Fig. 7 it can be seen that slight differences from MCNP6 results occurred in the core (0–60 cm) and RPV regions (100–125 cm) in groups 46 and 47 while the STRAUM results with different multi-group cross sections give good agreements. It is considered that the discrepancies from MCNP6 results would be caused due to the BUGLE-96 group structure, which has a relatively loose structure at low energy. In the case of gamma, there were considerable statistical errors in the MCNP results in the concrete region (125–200 cm), but the STRAUM results show good agreements with the MCNP calculation in the overall region as shown in Fig. 8

4.2. Numerical test problem 1

A simple box problem was designed to test the Krylov subspace methods implemented in the STRAUM code. Fig. 9 shows the layout of the simple box problem. As shown in the figure, a small box is located inside a large box. The macroscopic cross sections for the inner and outer boxes were taken from the seven group cross sections of the moderator and the guide tube of the C5G7 problem [23], respectively. The cross sections consider only isotropic scattering. The macroscopic scattering cross sections for the groups 5, 6, and 7 are coupled through up-scattering. As described above, in the STRAUM code, multi-group coupling iteration was performed by grouping groups 4, 5, 6, and 7 into one group chunk, and the groups above the chunk were sequentially calculated by applying conventional GS iteration in energy group. The number of tetrahedrons in the simple box problem is 19,671, and the vacuum boundary condition was applied to all external faces. the convergence criteria was set to be 10^{-7} of relative L_2 norm of the residual and $\vec{b}$ vectors. The calculation was performed using an Intel i5-9600KF CPU (3.70 GHz).

First, STRAUM using the Richardson, GMRES(30), and BiCGSTAB methods without a preconditioner were applied to this simple box problem with the sweep operation parallelized using 4 threads with Taskflow for each direction. The computing times for the different quadrature sets and group chunks are listed in Table 3. The time elapsed for the $\vec{b}$ vector setup to construct Eq. (11) is all equal regardless of the used Krylov subspace methods, and so the only average elapsed times for the parts in which three methods are performed are compared in Table 3. The elapsed times in Table 3 are the time difference between the start and the end of each iteration routine, and it includes the times for all the operations such as sweep operations and the operations related to vector algebra and memory copy. As a results of the calculation, it can be seen that GMRES(30) and BiCGSTAB takes substantially less computation times than Richardson for all the group chunks except for the first group chunk with GMRES(30). In particular, the most significant savings in computing time and number of sweeps for GMRES(30) and BiCGSTAB are observed in the last group chunk having up-scatterings, which shows that the Krylov subspace methods are much effective than the Richardson method for the chunk having up-scattering. Also, it is observed that the order of angular quadrature set does not affect the number of sweep operations and the speedup of computing time slightly increases as the order of angular quadrature set. For example, for 5x5 angular quadrature set, GMRES(30) and BiCGSTAB show the speedups of 6.9 and 10.0, respectively, compared with the Richardson iteration.

Table 4 compares the elapsed times when the DSA preconditioner for the single group chunks and the DiagDSA preconditioner for the last group chunk are applied. In the STRAUM code, the DSA preconditioner is explicitly performed using the matrix class of the Eigen library, so it goes through the process of setting up the DSA

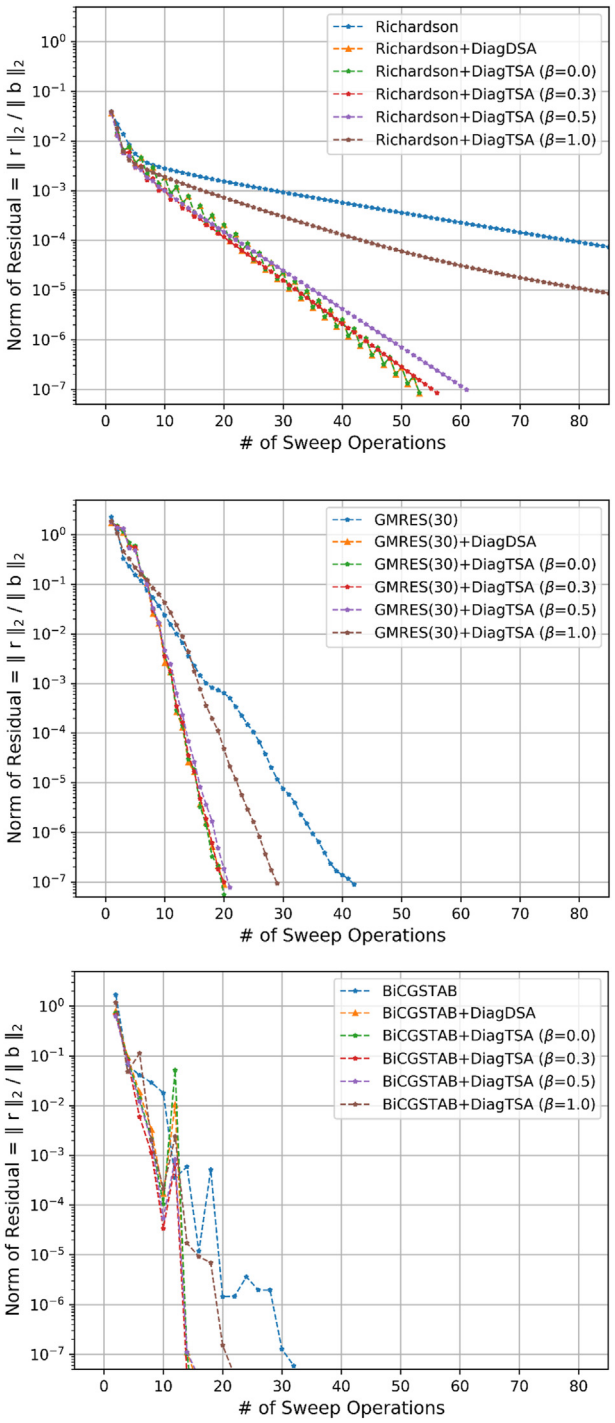

Fig. 10. L_2 norm of residual in the 4th group chunk according to the preconditioner; (top) Richardson, (middle) GMRES(30), (bottom) BiCGSTAB.

matrix before starting the iteration routine. The DSA setup consists of the calculation of the matrix elements, the setup of a sparse matrix, and the application of the preconditioner. The solution of the discretized diffusion equation was calculated using the BiCG-STAB solver with the diagonal preconditioner given in the Eigen library, and the convergence criteria was set as 10^{-5} in the relative L_2 norm of the residual and $\overrightarrow{b}$ vectors. As shown in Table 4, the DSA setup time is relatively small compared with the ones for the sweep operations and solving diffusion equation. The preconditioner does

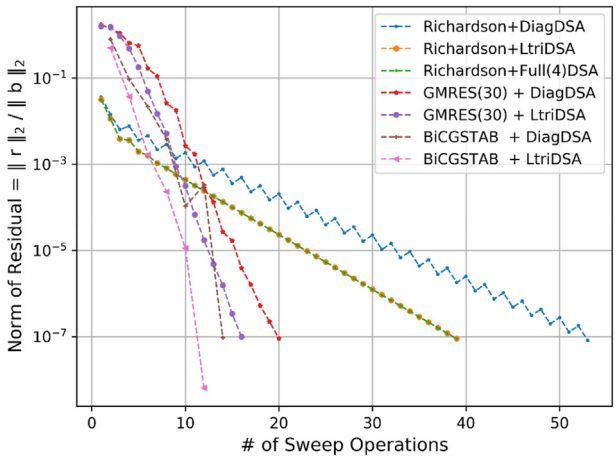

Fig. 11. Comparison of the L_2 norm of residuals for Krylov subspace methods with different preconditioners in 4th group chunk.

not give the computing time savings for the first group chunk because the first group chunk takes only small number of sweeps while the preconditioner gives considerably computing time savings for the other group chunks. In particular, the preconditioner in the last group chunk gives the largest reductions both in the number of sweepings and in computing time for all the considered methods. The computing time speedup by using preconditioner was the largest in the Richardson method where the Richardson method is also used in the last chunk. For example, the DiagDSA preconditioner for the Richardson method reduces the computing time by a factor of ~5.0 in the last group chunk. For the all group chunks, the BiCGSTAB method with the preconditioner showed the shortest computing time and the smallest number of sweeping. For the last group chunk, the BiCGSTAB method reduces the computing time by a factor of 18.0 in comparison with the Richardson method without a preconditioner.

Table 5 shows the elapsed times when the diagonal within-group block TSA preconditioner is applied. The TSA calculation was performed by the BiCGSTAB method using the 2x2 quadrature set considering only isotropic scattering, while the high-order transport calculation was performed using the 4x4 quadrature set. The various β values of 0.0, 0.3, 0.5, and 1.0 were considered to show the effects of this parameter on the computing time. Table 5 shows that the DiagTSA preconditioner gives the best performances with $\beta = 1.0$ for all the group chunks and all the methods due to the smallest computing time in the TSA calculations. In particular, it is noted that the DiagTSA with $\beta = 1.0$ preconditioner reduced computing time for all iterative method exccpt for the first group chunk but its performances worse than the ones with the DiagDSA preconditioner.

Fig. 10 shows the changes of the L_2 norm of residual normalized to $\overrightarrow{b}_2$ for the 4th group chunk as iteration for the Richardson, GMRES(30), and BiCGSTAB methods with the DiagDSA and DiagTSA preconditioners. As expected, the DiagTSA preconditioner shows better convergences for lower β values but the computing time was the shortest for $\beta = 1.0$. The DiagTSA method with $\beta = 0.0$ and 0.3 showed very similar convergence to the DiagDSA one for the Richardson and GMRES(30) methods. From Fig. 10, it can be observed that the BiCGSTAB method reduced the L_2 norm of residual with some fluctuations.

Fig. 11 compares the changes of the L_2 norm of residuals for the different iteration methods having the DiagDSA and LtriDSA preconditioners for the last group chunk. In the figure, "Full(4)DSA" refers to a multi-group DSA preconditioner in which 4 outer

Table 6
Elapsed time (seconds) of multi-group coupling iteration of 4th group chunk with different preconditioners using a 4x4 quadrature set.

	Richardson			GMRES(30)			BiCGSTAB		
	Time for Sweeps	Time for Precond.	Number of Iterations	Time for Sweeps	Time for Precond.	Number of Iterations	Time for Sweeps	Time for Precond.	Number of Iterations
DiagDSA	74.6	12.1	53	29.0	5.0	20	20.2	3.2	7 (14[a)])
LtriDSA	58.7	9.1	39	21.6	3.3	16	17.9	2.5	6 (12)
DiagTSA (β = 1.0)	272.4	89.9	188	38.8	13.8	29	30.1	10.4	11 (22)

[a)] Number of sweep operations.

Table 7
Comparison of total elapsed times (seconds) for the Krylov subspace methods using different preconditioners (4x4 angular quadrature set).

Iteration Method		Total Elapsed Time	Number of Iterations	Speedup to the Richardson method without preconditioner
Richardson	None	459.2	352	1.0
	DiagDSA	106.7	75	4.3
	LtriDSA	85.8	61	5.3
	DiagTSA (β =0.0)	498.7	67	0.9
	DiagTSA (β =0.3)	499.3	72	0.9
	DiagTSA (β =1.0)	385.4	215	1.2
GMRES(30)	None	85.7	70	5.4
	DiagDSA	48.4	41	9.5
	LtriDSA	43.4	37	10.6
	DiagTSA (β =0.0)	198.7	37	2.3
	DiagTSA (β =0.3)	194.3	38	2.4
	DiagTSA (β =1.0)	70.0	50	6.6
BiCGSTAB	None	60.0	26 (52[a)])	7.7
	DiagDSA	38.8	15 (30)	11.8
	LtriDSA	34.1	14 (28)	13.5
	DiagTSA (β =0.0)	132.0	13 (26)	3.5
	DiagTSA (β =0.3)	118.6	13 (26)	3.9
	DiagTSA (β =1.0)	53.2	18 (36)	8.6

[a)] Number of sweep operations.

iterations are applied to the DSA operations over the last group chunk. Fig. 11 shows that the Richardson method with the DiagDSA preconditioner gives some fluctuations in the residual change while the Richardson method with LtriDSA does not give the fluctuations and faster convergence than the DiagDSA one. Also, it is noted that the Full(4)DSA preconditioner shows similar convergence to the LtriDSA one even if it takes more number of energy group sweepings. Fig. 11 also shows that the GMRES(30) and BiCGSTAB method with the DiagDSA and LtriDSA preconditioners give much better convergences than the Richardson method. From this analysis, it was considered that the LtriDSA preconditioner was better than the DiagDSA one regardless of the iteration method in terms of computing time and convergence. The computing times taken for the 4th group chunk are compared in Table 6. The comparison of null 3 and 6 shows that the GMRES(30) and BiCGSTAB methods with the LtriDSA preconditioner and 4x4 quadrature reduce the computing time by factors of 16.9 and 20.8, respectively, in comparison with the Richardson method without a preconditioner for the last group chunk. Also, it is noted that the Richardson, GMRES(30), and BiCGSTAB methods with the LtriDSA preconditioner reduce the computing time by factors of 6.2, 2.6, and 2.2, respectively, in comparison with their unpreconditioned methods.

Table 7 shows the total elapsed times for the considered three Krylov subspace methods using different preconditioners and 4x4 angular quadrature set. The last column of the table shows the ratio of the total elapsed time of each method to that of the Richardson method without a preconditioner. Table 7 shows that the BiCGSTAB method had shorter computing times than the Richardson and GMRES(30) methods regardless of the preconditioner type. It is noted that the Richardson method with DiagTSA using $\beta = 0.0$ and 0.3 reduces the number of sweeps but does not reduce the computing time due to the long computing time in solving the TSA equations. BiCGSTAB with LtriDSA preconditioner took the shortest computation time and showed the speedup of 13.5 compared to the Richardson method without preconditioner.

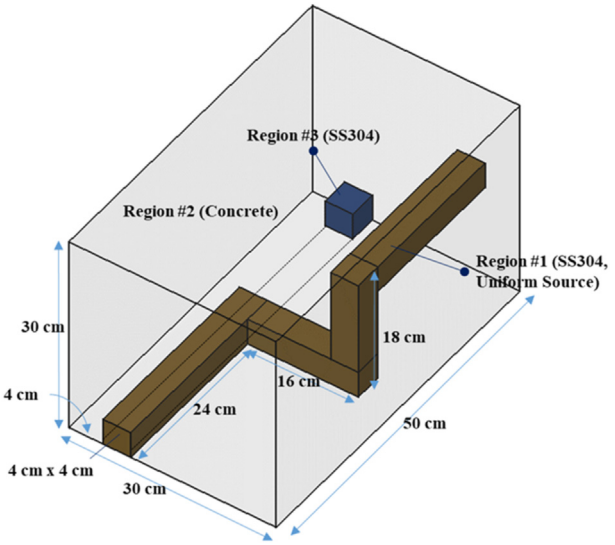

Fig. 12. Kobayashi-like problem layout.

4.3. Numerical test problem 2

The next problem for numerical test was devised by modifying

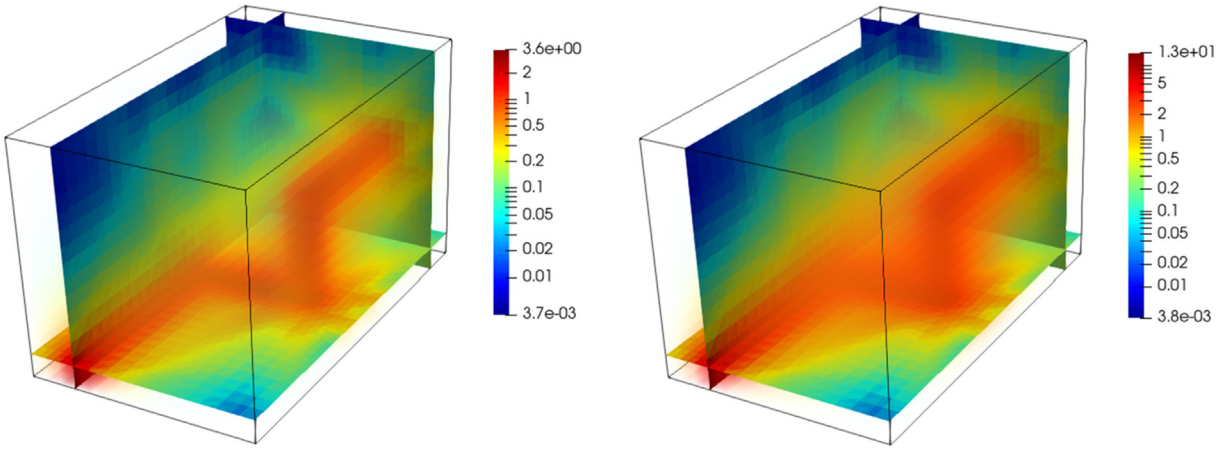

Fig. 13. Neutron flux distributions for group 1 (left) and group 10 (right) for the Kobayashi-like problem.

Table 8
Calculation conditions for the Kobayashi-like problem.

Parameter	Contents
# of Tetrahedrons	33,750
Quadrature Set	4 polar x 4 azimuthal Gauss-Chebyshev Quadrature per Octant
Cross Section	Neutron-gamma coupled cross section with 27n19g structure processed by MATXST based on ENDF/B-VIII.0
Anisotropic Order	P_3
Thermal Upscattering	Not considered
Convergence Criteria	Ratio of L_2 norm of residual to $\vec{b}_2 < 10^{-7}$
DSA & TSA Convergence Criteria	Ratio of L_2 norm of residual to $\vec{b}_2 < 10^{-5}$

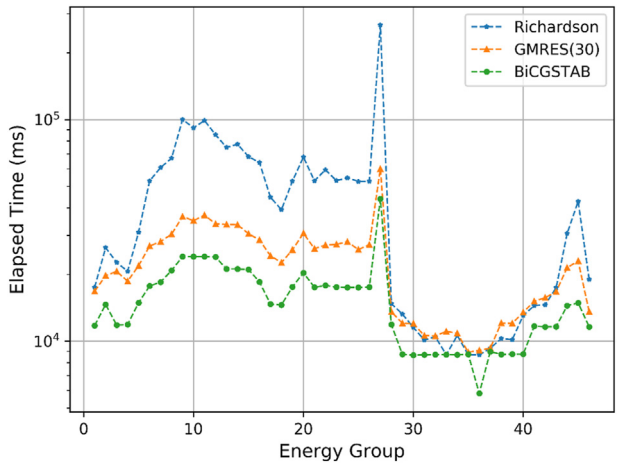

Fig. 14. Groupwise elapsed time for Kobayashi-like problem without preconditioner using a 4x4 quadrature set.

the Kobayashi's benchmark problem, which was designed for verifying transport codes with one-group cross sections [24]. The test problem is defined as a 30 cm × 50 cm x 30 cm hexahedron divided into 3 regions. The reflective condition was applied to the bottom surface (i.e., Z-surface), and the vacuum condition was applied to the remaining boundary surfaces. This Kobayashi-like problem uses the cross sections of realistic materials considering the scattering anisotropy up to the P_3 order. The cross sections of the MATXS format having a 27n19g structure were processed with MATXST and they have no up-scatterings. The first region (region 1) of SS304 representing a zigzagged pipe has an isotropic and uniform neutron source of 1.0 neutrons/cm^{3}sec only for the first ten energy groups (groups 1–10). The second region (region 2) occupying most of the problem is filled with concrete. The third region (region 3) is defined as a 4 cm × 4 cm x 4 cm cube of SS304. Fig. 12 shows the layout of the problem and Fig. 13 shows the neutron flux distributions for the groups 1 and 10. The parameters for this problem are described in Table 8. The calculation was performed using an Intel i5-9600KF CPU (3.70 GHz) with 4 threads for parallel computing for sweeping.

Fig. 14 shows the groupwise elapsed time of the Richardson, GMRES(30), and BiCGSTAB methods without preconditioner. As can be seen from the figure, BiCGSTAB had the shortest elapsed time and GMRES(30) took slightly longer computing time than BiCGSTAB, but the computing time of GMRES(30) significantly shorter than the Richardson method. However, in the high-energy groups of neutrons and most of the gamma groups, the speedups by the Krylov method are not significant because the Richardson method also shows the effective convergence. Fig. 15 shows the groupwise speedups in computing time for the Richardson, GMRES(30), and BiCGSTAB methods having different preconditioners relative to the corresponding methods without preconditioner. From the figures, it is shown that DSA was the most effective preconditioner, followed by TSA with $\beta = 1.0$, irrespective of the Krylov subspace methods. The TSA preconditioners with $\beta = 0.0$ and $\beta = 0.3$ take longer times in all energy groups, so it is expected that there will be some improvements in speedup if a higher quadrature set than 4x4 quadrature set is used. The high energy neutron groups and most of the gamma groups were converged rather quickly even without preconditioner.

Table 9 shows the total elapsed times for the considered Krylov subspace methods with different preconditioners. The fastest method was the BiCGSTAB method with the DSA preconditioner, followed by the BiCGSTAB without preconditioner and the next efficient method was the Richardson method with the DSA

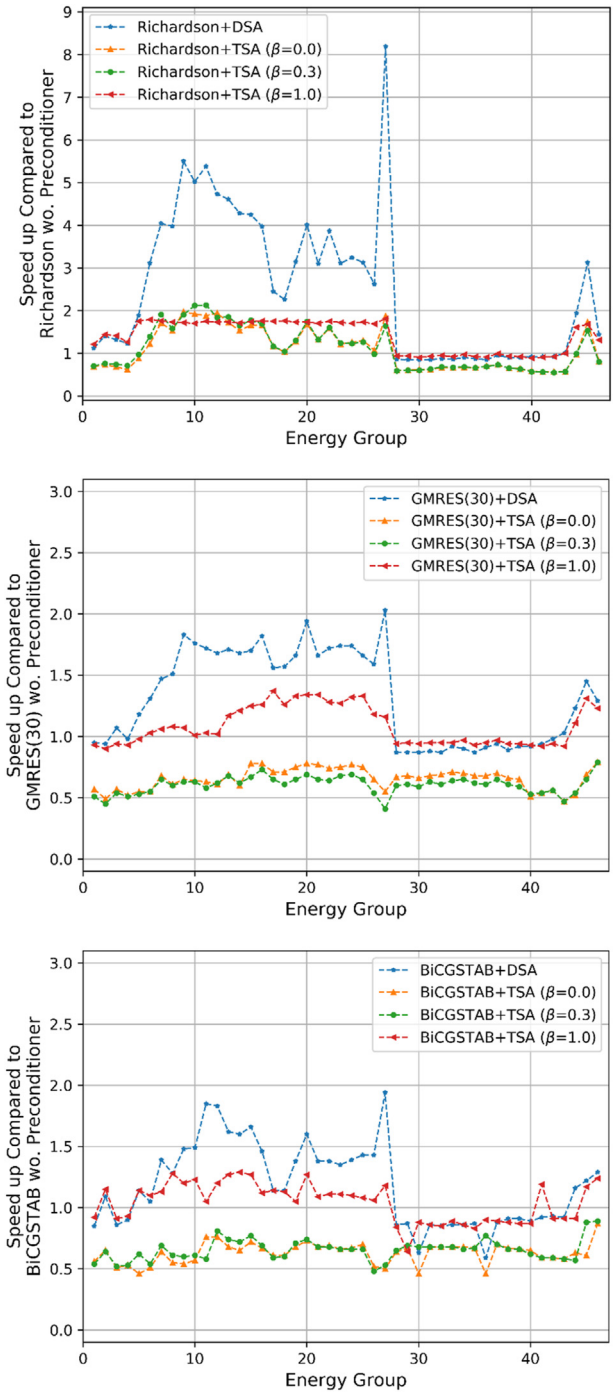

Fig. 15. Groupwise speedups for different preconditioners; (top) Richardson, (middle) GMRES(30), and (bottom) BiCGSTAB.

preconditioner. They reduce computing time by factors of 3.5, 2.9, and 2.8, respectively. The GMRES(30) with the DSA preconditioner showed a comparable speedup to the Richardson method with the DSA preconditioner.

4.4. Numerical test problem 3

The last test problem is a reactor pressure vessel (RPV) problem which was designed to test the convergence of the Krylov subspace methods in a more practical problem of reactor scale. The fuel assembly region was filled with ZIRLO, while the structural regions of the baffle, barrel, and vessel were defined as SS304. The neutron source of 1.0 $cm^{-3}sec^{-1}$ was assigned in the highest 10 groups among 47 groups. Fig. 16 shows the layout of the RPV problem, and Fig. 17 shows the tetrahedral meshes comprised of total 177,575 elements. The reflective condition was applied to three internal sides of the reactor while the vacuum condition was applied to the rest of the boundary surfaces. The neutron cross sections considering scattering anisotropy up to P_3 were prepared with BUGLE-96 group structure where the neutron energy groups span the first 47 groups and the last six groups (i.e., 42, -43, -44, -45, -46, and -47 groups) are coupled through upscattering. The last six groups are considered in a single group chunk. The parameters represent the calculation conditions are described in Table 10. The calculation was performed using an Intel i7-11700K CPU (3.60 GHz) with 5 threads for parallel computing for sweeping.

Fig. 18 shows the neutron flux distribution of group-1 of the RPV problem. Fig. 19 compares the changes of the L_2 norm of residual as iteration for the Krylov subspace methods with the DiagDSA and LtriDSA preconditioners for the last group chunk having upscattering. For this problem, the Richardson iterations with the DiagDSA and LtriDSA for the last group chunk did not converge. Fig. 19 shows the convergence of the Richardson method was particularly slow in the last group chunk and it took more than 1000 iterations to converge. On the other hand, it was observed that the residuals were efficiently reduced by the GMRES(30) and BiCGSTAB methods both with and without preconditioners. Although BiCGSTAB showed some fluctuations in residuals, its convergence was faster than that of GMRES(30). For both the GMRES(30) and BiCGSTAB methods, LtriDSA was more effective in computing time than DiagDSA for this problem.

Table 11 shows the elapsed times for the last group chunk having upscattering for the Krylov subspace methods with the DiagDSA and LtriDSA preconditioners. When the DiagDSA and LtriDSA preconditioners were applied, the time required for preconditioner was less than 5% of the time required for sweeping operation. The BiCGSTAB methods with the DiaDSA and LtriDSA preconditioners reduced the computing time for the last group chunk by factors of 16.4 and 21.3, respectively.

Table 12 shows the total elapsed times to solve the RPV problem. The last column of the table represents the ratio of the total elapsed time of each method to that of the Richardson method without preconditioner. For each Krylov subspace method, the DSA preconditioner were applied for all the group chunks except for the last group chunk where the specified precondition was applied. The results show that BiCGSTAB was more efficient in the RPV problem than GMRES(30), irrespective of the preconditioner type and the LtriDSA preconditioner showed slightly better performance than the DiagDSA one. The BiCGSTAB methods with DiagDSA and LtriDSA preconditioners gave the speedups of 10.2 and 11.5, respectively while the GMRES(30) ones gave slightly smaller speedups of 7.7 and 8.0 for the DiagDSA and LtriDSA preconditioners, respectively.

5. Summary and conclusion

In this work, a new multi-group discrete ordinates transport code system STRAUM-MATXST for complicated geometrical problems was introduced and its development status was reported. Currently, the STRAUM code uses tetrahedral meshes which are generated using Gmsh after the geometrical modeling with CAD softwares followed by a post processing the mesh files using an in-house program to use them in STRAUM. STRAUM uses the LDEM-SCB(1) discretization as the main transport solution option in which the coupling equations are derived using four sub-cell

Table 9
Total elapsed time (seconds) according to iteration method of Kobayashi-like problem using a 4x4 quadrature set.

Iteration Method	$\overrightarrow{b}$ setup (A)	Sweeping (B)	Number of Iterations	Precond. (C)	Etc. (D)	Total = A + B + C + D
Richardson	68.7	1,957.4	1332	-	6.0	2,032.0
Richardson + DSA	69.5	592.1	403	60.7	2.0	724.3
Richardson + TSA($\beta = 0.0$)	69.2	592.6	403	924.6	1.8	1,588.2
Richardson + TSA($\beta = 0.3$)	69.8	604.1	407	903.1	1.8	1,578.8
Richardson + TSA($\beta = 1.0$)	68.1	1,081.9	744	126.3	2.8	1,279.1
GMRES(30)	69.7	929.7	622	-	26.7	1,026.1
GMRES(30)+DSA	69.4	601.5	410	61.8	14.0	746.7
GMRES(30)+TSA($\beta = 0.0$)	70.1	608.7	408	905.4	13.8	1,597.9
GMRES(30)+TSA($\beta = 0.3$)	76.8	668.2	412	976.7	15.8	1,737.5
GMRES(30)+TSA($\beta = 1.0$)	71.4	757.7	503	84.1	19.8	932.9
BiCGSTAB	68.7	631.2	216 (432[a)])	-	5.5	705.4
BiCGSTAB + DSA	69.7	458.1	156 (312)	45.7	3.8	577.3
BiCGSTAB + TSA($\beta = 0.0$)	69.1	463.2	159 (318)	607.5	3.7	1,143.5
BiCGSTAB + TSA($\beta = 0.3$)	70.8	459.5	155 (310)	571.4	3.7	1,105.3
BiCGSTAB + TSA($\beta = 1.0$)	70.7	531.5	179 (358)	55.5	4.2	661.8

[a)] Number of sweep operations.

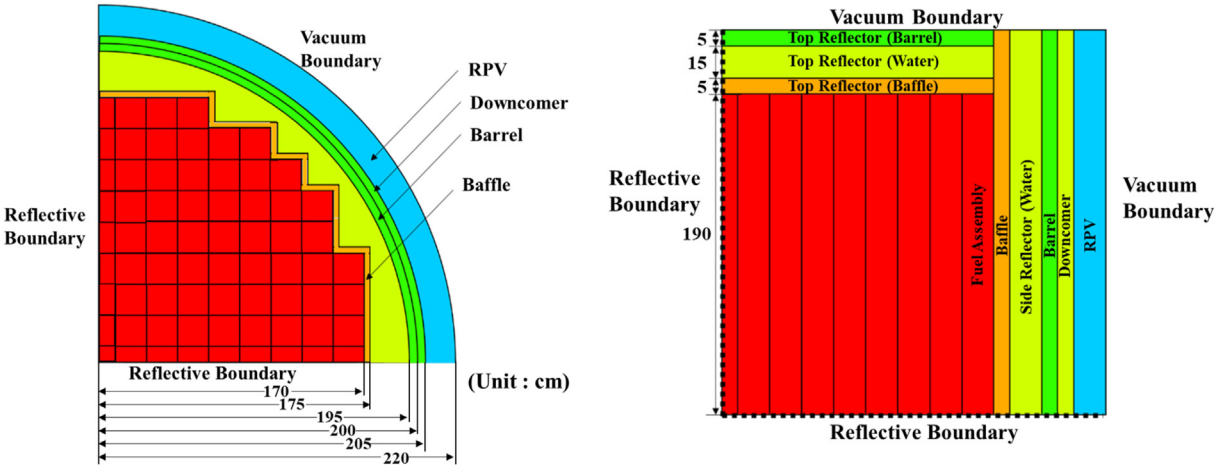

Fig. 16. Reactor pressure vessel problem layout.

balance equations and discontinuous flux expansion in each tetrahedral mesh. The MATXST code prepares the neutron-gamma coupled multi-group cross sections for STRAUM transport calculations after processing the MATXS format library generated with NJOY for considering transport corrections and resonance self-shielding effect with the Bondarenko iteration. In particular, the gamma production matrices are processed using the self-shielding factors to consider their self-shielding effect in MATXST code.

The special emphasis was given on the application of the Krylov subspace methods (i.e., BiCGSTAB and GMRES(m)) with DSA and β-TSA preconditioners for the within-group source iteration. STRAUM used the DSA equations for LDEM-SCB(1) transport discretization were derived in a consistent way from the continuous diffusion equation to the LDEM-SCB(1) discretization and the DSA equations were solved using BiCGSTAB method with the Eigen library after setting up the whole matrix considering its sparsity. . In addition, the Krylov subspace methods were extended to effectively solve multi-group coupled iterations having up-scattering with new multi-group block DSA and β-TSA preconditioners. Specifically, the multi-group block tri-diagonal DSA preconditioners as well as the simple block diagonal DSA and β-TSA ones were suggested and tested coupled with the Krylov subspace methods for several problems.

The verification of the MATXST code was done by comparing the multi-group cross sections generated with MATXST and TRANSX2 and by checking if the multi-group cross sections using different transport correction methods converge as the scattering anisotropy increases for a simple 1D shielding problem. Then, the validation of

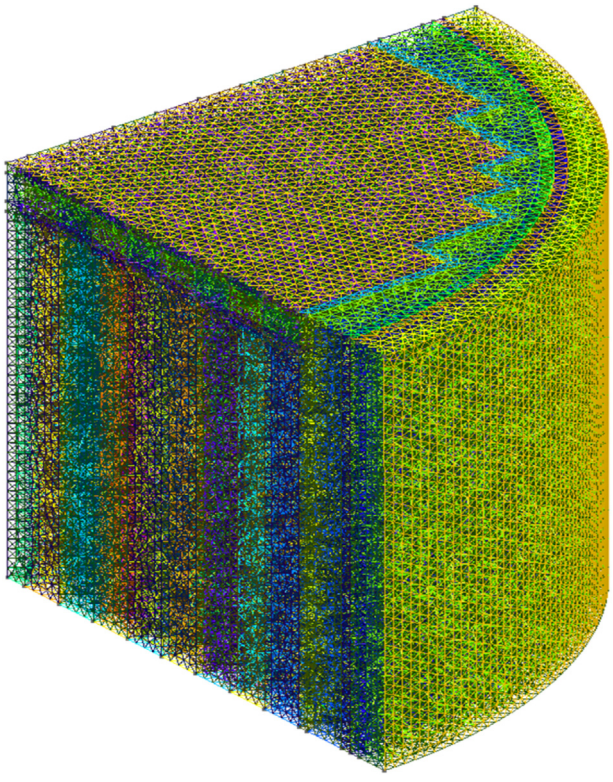

Fig. 17. Tetrahedron mesh for RPV problem generated using Gmsh.

Table 10
Calculation conditions for reactor pressure vessel problem.

Parameter	Contents
Number of Tetrahedrons	177,575
Quadrature Set	4 polar x 4 azimuthal Gauss-Chebyshev Quadrature per Octant
Cross Section	47 group cross section with BUGLE-96 structure processed by MATXST based on ENDF/B-VIII.0
Anisotropic Order	P_3
Thermal Upscattering	From 42 to 47
Convergence Criteria	Ratio of L_2 norm of residual to $\vec{b}_2 < 10^{-9}$
DSA & TSA Convergence Criteria	Ratio of L_2 norm of residual to $\vec{b}_2 < 10^{-5}$

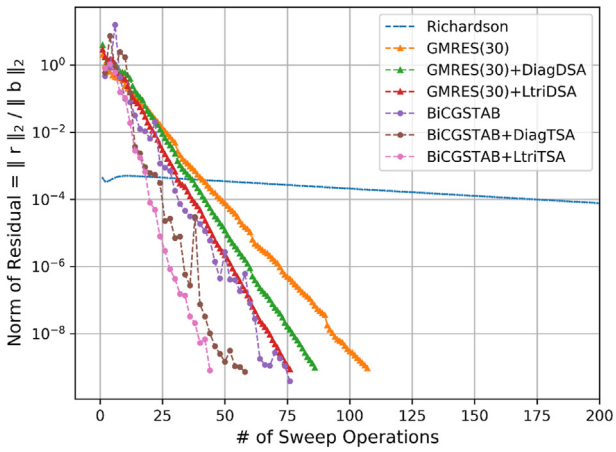

Fig. 19. L_2 norm of residual in multi-group coupling group chunk according to iteration method of the RPV problem.

the STRAUM and MATXST codes was done by comparing the multi-group flux distributions obtained with the STRAUM and MATXST codes and the continuous MCNP transport calculations for the 1D shielding problem. To check the performances of the Krylov sub-space methods in STRAUM, the detailed numerical tests were performed to one simple 3D shielding problem and to one realistic 3D reactor shielding problem containing the pressure vessel and concrete shield. From the numerical tests, it was shown that the BiCGSTAB and GMRES(m) methods were very effective for most groups and the DSA preconditioning both for the Richardson, BiCGSTAB, and GMRES(m) methods improves the convergences and reduces computing time in comparison with the conventional Richardson method for the within-group source iteration. In particular, BiCGSTAB showed the better speedups than GMRES(m) irrespective of the preconditioners and TSA preconditioner was much less effective than the DSA one for all the cases. It was noted that the TSA preconditioner does not give any computing time saving for the within-group source iteration for the cases taking large computing time portion in TSA calculation. For the energy group chunk having up-scattering, the GMRES(30) and BiCGSTAB methods with the suggested block preconditioners were very effective both in reducing the number of iterations and computing time in comparison with the Richardson iteration. For example, the BiCGSTAB and GMRES(30) methods with LtriDSA preconditioner for the last group chunk having up-scattering reduced the computing time by the factors of 21.3 and 16.4, respectively in comparison with the Richardson iteration for the realistic 3D reactor shielding problem. In the future, the parallel computing capability will be upgraded and the code system will be extended to include the activation calculation capability with an in-house depletion code.

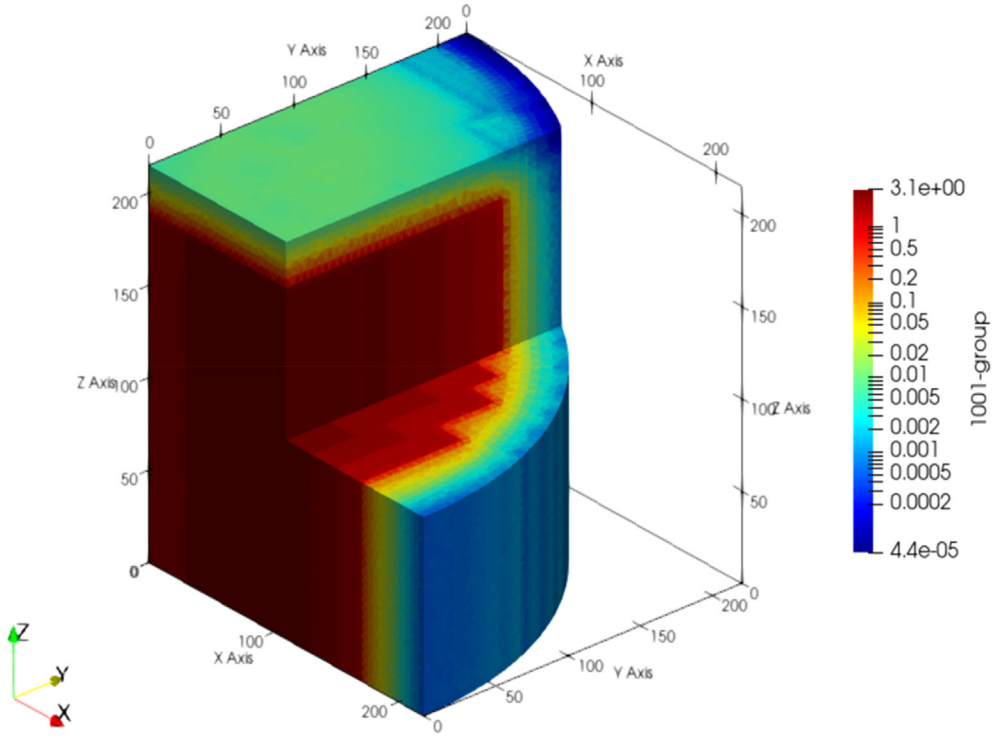

Fig. 18. Neutron flux distribution in the first group of RPV problem.

Table 11
Elapsed times in the last group chunk for different Krylov subspace methods using preconditioners of RPV problem using a 4x4 quadrature set (in seconds).

Iteration Method	$\vec{b}$ setup (A)	Sweeping (B)	Number of Iterations	Precond. (C)	Etc. (D)	Total = A + B + C + D
Richardson	48.4	47312.7	1001	-	6683.4	54044.5
Richardson + DiagDSA	Not Converged					
Richardson + LtriDSA	Not Converged					
GMRES(30)	48.0	4888.4	108	-	170.8	5107.2
GMRES(30)+DiagDSA	47.0	3921.4	87	148.3	132.6	4249.3
GMRES(30)+LtriDSA	47.8	3533.8	77	124.8	109.4	3815.8
BiCGSTAB	51.3	3671.7	38 (76[a)])	-	24.4	3747.4
BiCGSTAB + DiagDSA	55.3	3143.1	29 (59)	88.7	14.9	3302.0
BiCGSTAB + LtriDSA	55.3	2399.5	22 (44)	65.6	13.5	2533.9

[a)] Number of sweep operations.

Table 12
Comparison of total elapsed time (seconds) of RPV problem according to iteration method when using a 4x4 quadrature set.

Iteration Method		Total Elapsed Time	Number of Iterations	Speedup to the Richardson method without preconditioner
Richardson	without Precond.	77254.7	3092	1.0
GMRES(30)	without Precond.	15753.3	947	4.9
	DiagDSA	10071.4	564	7.7
	LtriDSA	9670.1	554	8.0
BiCGSTAB	without Precond.	9846.9	290 (580[a)])	7.8
	DiagDSA	7562.9	174 (348)	10.2
	LtriDSA	6724.1	167 (334)	11.5

[a)] Number of sweep operations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the NRF (National Research Foundation of Korea) through Project No. NRF-2019M2D2A1A02057890 and the Nuclear Safety Research Program through the Korea Foundation Of Nuclear Safety (KoFONS) using the financial resource granted by the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea. (No. 2101075)

References

[1] T.A. Wareing, J.M. McGhee, J.E. Morel, ATTILA: a three-dimensional, unstructured tetrahedral mesh discrete ordinates transport code, Trans. Am. Nucl. Soc. 75 (1996).
[2] R.L. Martz, The MCNP6 book on unstructured mesh geometry: user's guide, Los Alamos Natl. Lab. (2014) 1–77. LA-UR-11-05668.
[3] T.D. Long, S.G. Hong, Implementation and Verification of adjoint neutron transport calculation in MUST code, in: Transactions of the Korea Nuclear Society Virtual Spring Meeting, Republic of Korea, 2020. July 9-10.
[4] T. Wareing, J.M. McGhee, J. Morel, S. Pautz, Discontinuous finite element SN methods on three-dimensional unstructured grids, Nucl. Sci. Eng. 138 (2001) 256–268, https://doi.org/10.13182/NSE138-256.
[5] S.G. Hong, Two subcell balance methods for solving the multigroup discrete ordinates transport equation with tetrahedral meshes, Nucl. Sci. Eng. 173 (2013), https://doi.org/10.13182/NSE11-38.
[6] H. Muhammad, S.G. Hong, A three-dimensional Fourier analysis of fine mesh rebalance acceleration of linear discontinuous sub-cell balance method for diffusion equation on tetrahedral meshes, Ann. Nucl. Energy 133 (2019) 145–153, https://doi.org/10.1016/J.ANUCENE.2019.05.020.
[7] Y. Azmy, E. Sartori, Nuclear Computational Science: a Century in Review, Springer, 2010.
[8] R.N. Slaybaugh, Vermaak, T.M. Evans, G.G. Davidson, P.P.H. Wilson, Multigrid in energy preconditioner for Krylov solvers, J. Comput. Phys. 242 (2013) 405–419.
[9] Christophe Geuzaine, Jean-Francois Remacle, Gmsh, Available at: http://gmsh.info/, 2020.
[10] M.H. Woo, T.D. Long, S.G. Hong, Development of multi-group cross section processing program for MUST unstructured discrete ordinate transport code, in: Transactions of the Korea Nuclear Society Virtual Autumn Meeting, Republic of Korea, 2020. December 17-18.
[11] T.W. Huang, D.L. Lin, C.X. Lin, Y. Lin, Taskflow: a lightweight parallel and heterogeneous task graph computing system, IEEE Trans. Parallel Distrib. Syst. 33 (2022), https://doi.org/10.1109/TPDS.2021.3104255.
[12] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986), https://doi.org/10.1137/0907058.
[13] H.A. van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (1992), https://doi.org/10.1137/0913035.
[14] H. Muhammad, S.G. Hong, Diffusion synthetic acceleration with the fine mesh rebalance of the subcell balance method with tetrahedral meshes for S_N transport calculations, Nucl. Eng. Technol. 52 (2020), https://doi.org/10.1016/j.net.2019.08.021.
[15] G. Guennebaud, B. Jacob, Eigen v3, others, Available at: https://eigen.tuxfamily.org/index.php?title=Main_Page, 2010.
[16] M.L. Adams, E.W. Larsen, Fast iterative methods for discrete-ordinates particle transport calculations, Prog. Nucl. Energy 40 (2002) 3–159, https://doi.org/10.1016/S0149-1970(01)00023-3.
[17] R. Macfarlane, D.W. Muir, R.M. Boicourt, I.A.C. Kahler, J.L. Conlin, The NJOY Nuclear Data Processing System, 2017, https://doi.org/10.2172/1338791. Version 2016, Los Alamos, NM (United States).
[18] R.E. Macfarlane, TRANSX 2: a code for interfacing MATXS cross-section libraries to nuclear transport codes, Los Alamos Natl. Lab. (1993). LA-12312-MS.
[19] C.J. Werner, MCNP users manual, Code Version 6.2, Los Alamos Natl. Lab. (2017). LA-UR-29981.
[20] D.A. Brown, M.B. Chadwick, et al., ENDF/B-VIII.0: the 8th major release of the nuclear reaction data library with CIELO-project cross sections, new standards and thermal scattering data, Nucl. Data Sheets 148 (2018) 1–142, https://doi.org/10.1016/j.nds.2018.02.001.
[21] D. Wiarda, M.E. Dunn, D.E. Peplow, T.M. Miller, H. Akkurt, Development and testing of ENDF/B-VI. 8 and ENDF/B-VII. 0 coupled neutron-gamma libraries for scale 6, Oak Ridge Natl. Lab. (2009). ORNL/TM-2008/047.
[22] J.E. White, D.T. Ingersoll, C.O. Slater, R.W. Roussin, BUGLE-96: a revised multigroup cross section library for lwr applications based on ENDF/B-VI release 3, Oak Ridge Natl. Lab. (1996). CONF-960415-37.
[23] E.E. Lewis, M.A. Smith, N. Tsoulfanidis, G. Palmiotti, T.A. Taiwo, R.N. Blomquist, Benchmark specification for Deterministic 2-D/3-D MOX fuel assembly transport calculations without spatial homogenization (C5G7 MOX), Nucl. Energy Agency, Organisation Econ. Co-operation Dev. 2001, NEA/NSC/DOC (2001) 4.
[24] K. Kobayashi, N. Sugimura, Y. Nagaya, 3D radiation transport benchmark problems and results for simple geometries with void region, Prog. Nucl. Energy 39 (2001) 119–144, https://doi.org/10.1016/S0149-1970(01)00007-5.