Annals of Nuclear Energy 38 (2011) 2742–2751

Contents lists available at SciVerse ScienceDirect

Annals of Nuclear Energy

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

An acceleration technique for 2D MOC based on Krylov subspace and domain decomposition methods

Hongbo Zhang, Hongchun Wu, Liangzhi Cao *

School of Nuclear Science and Technology, Xi'an Jiaotong University, Xi'an Shaanxi 710049, China

ARTICLE INFO

Article history:
Received 23 April 2011
Received in revised form 12 August 2011
Accepted 16 August 2011
Available online 24 September 2011

Keywords:
MOC
Neutron transport
Acceleration technique
Krylov subspace method
Domain decomposition method
Parallelization

ABSTRACT

The method of characteristics (MOC) has great geometrical flexibility but poor computational efficiency in neutron transport calculations. The generalized minimal residual (GMRES) method, a type of Krylov subspace method, is utilized to accelerate a 2D generalized geometry characteristics solver AutoMOC. In this technique, a form of linear algebraic equation system for angular flux moments and boundary fluxes is derived to replace the conventional characteristics sweep (i.e. inner iteration) scheme, and then the GMRES method is implemented as an efficient linear system solver. This acceleration method is proved to be reliable in theory and simple for implementation. Furthermore, as introducing no restriction in geometry treatment, it is suitable for acceleration of an arbitrary geometry MOC solver. However, it is observed that the speedup decreases when the matrix becomes larger. The spatial domain decomposition method and multiprocessing parallel technology are then employed to overcome the problem. The calculation domain is partitioned into several sub-domains. For each of them, a smaller matrix is established and solved by GMRES; and the adjacent sub-domains are coupled by "inner-edges", where the trajectory mismatches are considered adequately. Moreover, a matched ray tracing system is developed on the basis of AutoCAD, which allows a user to define the sub-domains on demand conveniently. Numerical results demonstrate that the acceleration techniques are efficient without loss of accuracy, even in the case of large-scale and strong scattering problems in complex geometries.

1. Introduction

In the process of reactor physical analysis, neutron transport equation often has to be solved accurately in very complicated geometries. Among various deterministic methods for neutron transport calculation, the method of characteristics (MOC) is the best candidate to treat strong geometrical heterogeneities (Askew, 1972). In this algorithm, sets of trajectories crossing the computational domain are generated in each of the discretized directions and intersect numerous arbitrary-shaped spatial meshes with flat or linear source approximations. Then the neutron transport equation is solved by sweeping the trajectories repeatedly to obtain the angular fluxes and mean scalar flux in every mesh.

Owing to the great geometrical flexibility by nature, MOC has been included as an important neutron transport solving module by many reactor analyzing softwares in recent years (Sanchez et al., 1988; Knott et al., 1995; Halsall, 1998). AutoMOC is one of the MOC programs (Chen et al., 2008). Its novelty consists in the use of the powerful functionality on engineering graphics and customizations offered by the computer aided design software AutoCAD in geometry processing and ray tracing. Distinguished from some other MOC solvers which utilize modular ray tracing technique (Halsall, 1980; Cho et al., 2008; Tang and Zhang, 2009), AutoMOC is based on the long characteristics technique. It generates characteristics rays in the entire problem domain and hardly imposes limitation on geometry.

However, AutoMOC encounters computational efficiency problems just like other MOC solvers do; hence an effective and geometry-flexible acceleration technique is urgently needed. The well-known acceleration technique coarse-mesh finite difference (CMFD) method has been applied to MOC solvers for large-scale calculations in reactors (Joo et al., 2002; Cho et al., 2008; Tang and Zhang, 2009) and favorable effects were gained. However, the method which is based on the finite difference format has the inherent drawback of geometrical applicability. The difficulty is overcome by the generalized coarse-mesh rebalance (GCMR) (Yamamoto, 2005) and the generalized coarse-mesh finite difference (GCMFD) method (Chai et al., 2010). Unfortunately, a factor which determines the convergence property cannot be given as *a priori*, which is a flaw in theory though it could be handled numerically as an expedient. Some other techniques such as the multigrid method (Grassi, 2007) have been proposed to accelerate MOC in

* Corresponding author. Tel.: +86 29 82663285; fax: +86 29 82667802.
E-mail address: caolz@mail.xjtu.edu.cn (L. Cao).

0306-4549/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.anucene.2011.08.015

generalized geometry. However, sometimes the acceleration effects may be weakened by the time-consuming solution of acceleration equations.

In this study, an acceleration technique based on the Krylov subspace methods for linear algebraic equation systems is conducted. It requires no additional acceleration step and avoids redundant computational effort. A form of linear system is constructed to replace the conventional characteristics sweep (i.e. inner iteration) scheme in MOC. The angular flux moments and boundary angular fluxes are involved in the linear systems instead of the scalar fluxes and boundary currents respectively (Dahmani et al., 2005), which makes the description of transport problems more precisely. Then the equation systems are solved by the generalized minimal residual (GMRES) method, which is a widely-used Krylov subspace method for solving systems with non-symmetric coefficient matrices efficiently. Comparing with another Krylov subspace method Lanczos algorithm (Santandrea and Sanchez, 2002), the GMRES method skips the symmetrization process and permits the potential use of several preconditioners. The geometry flexibility of this technique is perfect due to the theoretical equivalence between the original MOC and the accelerated one. Moreover, it is reliable in theory and simple for implementation.

However, it is observed that the speedup decreases when the matrix becomes larger in the Krylov subspace acceleration method. The main reason is that the time spent on both the construction and solution of the matrix increases sharply as the matrix grows large. This problem causes the inability of the Krylov subspace acceleration method in dealing with multi-assembly-level or core-level computations.

The domain decomposition (DD) method is a powerful tool for large-scale scientific computations. Its principle is "divide-and-conquer". In this algorithm, the total computational domain is divided into several overlapping or non-overlapping sub-domains, each of which is coupled with its own adjacent sub-domains. Calculations are carried out separately in each sub-domain and then communications occur between adjacent ones. It has been proved that the above iteration process will finally converge to the true solution of the original problem if only the coupling conditions are appropriate (Saad, 1996). As a result, a large-scale problem is successfully converted into several smaller ones by the DD method. In addition, calculations in sub-domains are independent to some degree; hence multiprocessing parallel technology could be utilized to improve the computational efficiency.

The DD method has been applied to some famous MOC solvers as a parallel computing scheme. An angular DD method is invoked in CRX (Lee et al., 2000) and GALAXY (Yamaji et al., 2010), and laudable parallel efficiency is obtained though it is not so economical on treatments of the white boundary condition. A spatial DD method is utilized in CHAPLET (Kosaka and Saji, 2000), which is an assembly modular ray tracing MOC solver instead of a 2D generalized geometry one. In our study, a spatial non-overlapping domain decomposition method, which is based on the 2D generalized geometry Krylov-accelerated MOC solver AutoMOC, is proposed to solve the aforementioned problem that the Krylov subspace acceleration method is incapable of large-scale computations. By partitioning the space domain into sub-domains, a burdensome matrix is converted into smaller ones, which are much easier to be constructed and solved. Adjacent sub-domains are coupled by "inner-edges", where the trajectory mismatches between adjacent sub-domains are considered adequately. Multiprocessing parallel technology is utilized after an adjustment on the conventional computation flow. Moreover, a matched ray tracing tool is developed on the basis of the original AutoMOC ray tracing system, which allows the user to define the sub-domains on demand conveniently.

The remainder of this paper is organized as follows. Section 2 exhibits the methodology of this work, including the mathematical derivation and details in implementation. Section 3 is dedicated to some numerical results which demonstrate the relative efficiency and accuracy compared with the original program AutoMOC. Finally we draw some conclusions and make a few suggestions in Section 4.

2. Theoretical model

Two parts exist in this section. The first part is for the Krylov subspace acceleration method, including the derivation of the MOC linear system and implementation details. The second part is about the DD method, which is composed of the theoretics, the geometry processing technique and the parallelization technique.

2.1. The Krylov subspace acceleration method

2.1.1. Basic equations of MOC

In the MOC calculation, the computational domain is partitioned into a number of regions, in each of which the source and cross-section are both assumed to be constants. Hence the time-independent multigroup neutron transport equation can be written in region i on track line segment k by omitting the energy group superscripts g:

$$\frac{d\psi_{i,k}(s,\vec{\Omega})}{ds}+\Sigma_{t,i}\psi_{i,k}(s,\vec{\Omega})=Q_{i,k}(\vec{\Omega}) \tag{1}$$

where s is the local coordinate along the track line. Thus the angular flux distribution along the track line can be obtained by the analytical solution of Eq. (1):

$$\psi_{i,k}(s,\vec{\Omega})=\psi_{i,k}^{in}(\vec{\Omega})\exp(-\Sigma_{t,i}s)+\frac{Q_{i,k}(\vec{\Omega})}{\Sigma_{t,i}}\left[1-\exp(-\Sigma_{t,i}s)\right] \tag{2}$$

where $\psi_{i,k}^{in}(\vec{\Omega})$ is the incident angular flux to region i along line segment k in direction Ω. The mean angular flux in region i along direction Ω is considered as

$$\overline{\psi}_i(\vec{\Omega})=\frac{\sum_k \overline{\psi}_{i,k}(\vec{\Omega})s_{i,k}\delta A_k}{V_i} \tag{3}$$

here, $\overline{\psi}_{i,k}$ is the mean angular flux along segment k, which can be obtained by integrating Eq. (2) along k; and V_i is the volume of region i, which can be approximately represented as $V_i\approx\sum_k s_{i,k}\delta A_k$, where the sum is over all the segments in region i along direction Ω, involving the segment length $s_{i,k}$ and its width δA_k. Thus Eq. (3) has the following form:

$$\overline{\psi}_i(\vec{\Omega})=\frac{\sum_k \delta A_k\left[Q_i^{iso}s_{i,k}+Q_i^{aniso}(\vec{\Omega})s_{i,k}+\left(\psi_{i,k}^{in}((\vec{\Omega}))-\psi_{i,k}^{out}(\vec{\Omega})\right)\right]}{\Sigma_{t,i}V_i} \tag{4}$$

where $\psi_{i,k}^{out}$ is the exiting flux along k from region i; Q_i^{iso} and Q_i^{aniso} are respectively the isotropic and anisotropic source in region i. Hence the mean scalar flux in region i can be written after obtaining the mean angular flux in each discretized direction:

$$\phi_i=\int_{4\pi}\overline{\psi}_i(\vec{\Omega})d(\vec{\Omega})\approx\sum_{m=1}^{M}\omega_m\overline{\psi}_i(\vec{\Omega}) \tag{5}$$

where M is the total number of discretized directions; and ω_m is the weight of direction m.

On the treatment of reflective boundary conditions in complex geometry, mismatches between incoming and outgoing directions are often encountered due to the variety of outer boundary. As the illustration in Fig. 1, $\Omega_{m'}$ is the specular reflection direction of Ω_m.

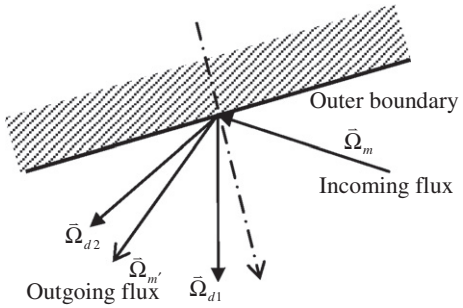

Fig. 1. Treatment of reflective B.C.

In theory, the incoming angular flux $\psi(\vec{\Omega}_m)$ equals the outgoing one $\psi(\vec{\Omega}_{m'})$ exactly. Unfortunately, in this case, $\vec{\Omega}_{m'}$ does not exist in the chosen angular discretization scheme, which causes the equality above to fail. To vanquish it, two adjacent directions of $\vec{\Omega}_{m'}$ (i.e. $\vec{\Omega}_{d1}$ and $\vec{\Omega}_{d2}$) are selected from the existing discretized directions and an angular interpolation is employed:

$$\psi(\vec{\Omega}_m) = \psi(\vec{\Omega}_{m'}) = C_m \cdot \psi(\vec{\Omega}_{d1}) + (1 - C_m) \cdot \psi(\vec{\Omega}_{d2}) \tag{6}$$

$$C_m = \frac{\theta_{d2} - \theta_{m'}}{\theta_{d2} - \theta_{d1}} \tag{7}$$

In the AutoMOC scheme, the outer boundary is subdivided into "edges" by the adjacent regions to this boundary. On each of the edges, the angular flux is independent of space variables. Considering the albedo boundary condition on edge s, Eq. (6) becomes

$$\psi^-_{s,m} = \beta\psi^+_{s,m'} = \beta C_m \cdot \psi^+_{s,d1} + \beta(1 - C_m) \cdot \psi^+_{s,d2} \tag{8}$$

where $\psi^-_{s,m}$ indicates the incoming angular flux in direction $\vec{\Omega}_m$ on edge s, and $\psi^+_{s,m'}$ is the outgoing one in direction $\vec{\Omega}_{m'}$; β is the albedo factor.

2.1.2. Linear algebraic equation system of MOC

The linear system of MOC, which will replace the conventional recurrent characteristics sweep in inner iterations, involves angular flux moments in each region and incident angular fluxes from each outer boundary edge. The angular flux moment $\phi_{n,l}$ is defined as

$$\begin{aligned}\phi_{n,l}(\vec{r}) &= \frac{1}{4\pi}\int_{4\pi} Y_{n,l}(\vec{\Omega})\psi(\vec{r},\vec{\Omega})d\vec{\Omega} \\ &\approx \frac{1}{4\pi}\sum_m w_m Y_{n,l}(\vec{\Omega}_m)\psi(\vec{r},\vec{\Omega})_m, \quad |l| \le n\end{aligned} \tag{9}$$

where $Y_{n,l}(\vec{\Omega}_m)$ is the spherical harmonics with the following definition:

$$Y_{n,l}(\vec{\Omega}) = \alpha_{n|l|}P_n^{|l|}(\mu) \times \begin{cases} \cos l\varphi & l \ge 0 \\ \sin|l|\varphi & l < 0 \end{cases} \tag{10}$$

where φ is the azimuthal angle of $\vec{\Omega}_m$ and μ is the cosine of the polar angle; $\alpha_{n|l|}$ is the normalization constant and $P_n^{|l|}(\mu)$ is the Legendre polynomial:

$$\alpha_{nl} = \sqrt{(2 - \delta_{l0})\frac{(n-l)!}{(n+l)!}} \tag{11}$$

$$P_n^l(\mu) = (1 - \mu^2)^{\frac{l}{2}}\frac{d^l P_n(\mu)}{d\mu^l} \tag{12}$$

Hence, the following expressions can be easily obtained:

$$Y_{0,0}(\vec{\Omega}) = 1; \quad Y_{1,0}(\vec{\Omega}) = \mu; \quad Y_{1,1}(\vec{\Omega}) = \sqrt{1-\mu^2}\cos\varphi; \quad Y_{1,-1}(\vec{\Omega}) = \sqrt{1-\mu^2}\sin\varphi$$

.

Since the linear approximation for anisotropic scattering source term is adopted in AutoMOC, only four orders of moments exist: $\phi_{0,0}$, $\phi_{1,-1}$, $\phi_{1,0}$ and $\phi_{1,1}$. According to Eqs. (9)–(12), among the four moments, $\phi_{0,0}$ is $\frac{1}{4\pi}$ times of scalar flux actually; $\phi_{1,0}$ is identically vanishing due to the symmetric angular flux distribution in a 2D problem, and it will be ignored hereafter. Thus, for convenience, the remaining unknowns $\phi_{0,0}$, $\phi_{1,-1}$ and $\phi_{1,1}$ will be denoted by ϕ_0, ϕ_1 and ϕ_2; and the corresponding spherical harmonic functions are denoted in the same way.

According to the definition of the angular flux moments, the source terms of energy group g in Eq. (4) can be represented in the forms of moments:

$$Q_i^{iso,g} = Q_i^{f,g} + Q_i^{s,iso,g} = \frac{\chi_g}{k_{eff}}\sum_{g'=1}^{G}(\nu\Sigma)_{f,g',i}\phi^{g'}_{0,i} + \sum_{g'=1}^{G}\Sigma_{s0,g'g,i}\phi^{g'}_{0,i} \tag{13}$$

$$Q_i^{aniso,g}(\vec{\Omega}) = Q_i^{s,aniso,g}(\vec{\Omega}) = 3\sum_{g'=1}^{G}\Sigma_{s1,g'g,i}\sum_{\rho=1}^{2}Y_\rho(\vec{\Omega})\phi^{g'}_{\rho,i} \tag{14}$$

where $Q_i^{f,g}$, $Q_i^{s,iso,g}$ and $Q_i^{s,aniso,g}(\vec{\Omega})$ denote the fission source, the isotropic scattering source and the anisotropic scattering source in region i respectively. Furthermore, each of the source term can be represented as the sum of out-of-group source and in-group source. For instance, $Q_i^{iso,g}$ is the sum of $Q_i^{iso,gg}$ and $Q_i^{iso,g'g}$; where $Q_i^{iso,gg}$ is the isotropic source contributed by group g itself, and $Q_i^{iso,g'g}$ is the isotropic source from the other groups. As the linear system to be derived is the alternative to inner iterations, the in-group source is unknown but the out-of-group source will be considered as a definite right hand side (RHS).

According to the definition above, a propagation equation is obtained by the successive use of Eq. (2):

$$\psi^{out}_{i,k}(\vec{\Omega}_m) = e^k_{1,i}\psi^-_k(\vec{\Omega}_m) + \sum_{\rho=0}^{2}\sum_{j=1}^{i}t^k_{\rho j,i}\phi_{\rho,j} + q_i^{k,g'g} \tag{15}$$

where $\psi^{out}_{i,k}(\vec{\Omega}_m)$ is the outgoing angular flux of region i along track line k in direction m; ψ^-_k denotes the incoming angular flux at the starting point of the line on a certain boundary; the second term on the RHS indicates the in-group source contributed by the upstream regions along track line k to region i; and $q_i^{k,g'g}$ means the out-of-group source from the upstream regions. Actually, $e^k_{1,i}$ is the exponential attenuation coefficient from the starting region to region i; and $t^k_{\rho j,i}$ represents the attenuation coefficient from region j to region i.

For the order P angular flux moment in a specified region I, Eq. (9) becomes

$$\phi_{P,I} = \frac{1}{4\pi}\sum_{m=1}^{M}w_m Y_P(\vec{\Omega}_m)\overline{\psi}_I(\vec{\Omega}_m) \tag{16}$$

By substituting Eq. (4) into Eq. (16), and distinguishing the out-of-group sources from the in-group ones, one can finally obtain the equation for angular flux moments after some mathematical implementations with the assistance of Eq. (15):

$$\left[a\phi_{P,I} + \sum_{\rho=0}^{2}\sum_{i_k}\tau\phi_{\rho,i_k}\right] + \sum_{k_I}\varepsilon\psi^-_{s,m} = Q_1^{g'g} \tag{17}$$

where the symbol i_k denotes a certain upstream region of I (including I itself) along a certain track line k_I which crosses region I. In this equation, the first term on the left hand side (LHS) stands for the Pth

order angular flux moment in region I; the second term represents the sum of the other moments in region I, and the moments in the upstream regions; the last one on the LHS indicates the sum of the incident angular fluxes from outer boundaries at the endpoints of each track line which crosses I, where the outer boundary edge s and the line direction m may vary with different track lines. The RHS $Q_1^{g'g}$ has a physical meaning of an out-of-group source. The coefficient a represents the capture probability in region I. τ and ε are the in-group net contribution from the upstream regions and outer boundary respectively.

The equation for outer boundary fluxes will be derived by the example of albedo conditions though the other common conditions are also available in this method. Since the MOC linear system involves the incoming angular fluxes instead of outgoing ones in outer boundary, a conversion from the outgoing angular fluxes to the incoming ones at the opposite endpoint of the same track line has to be made by the usage of Eq. (15). After the conversion and some mathematical transformation, Eq. (8) becomes the equation for incident angular fluxes:

$$\left[\sum_{\rho=0}^{2}\sum_{i_{d1}=1}^{I_s} t_1\phi_{\rho,i_{d1}} + \sum_{\rho=0}^{2}\sum_{i_{d2}=1}^{I_s} t_2\phi_{\rho,i_{d2}}\right] + \left[\bar{\psi}^{-}_{s,m} + \sum_{s1} e_1\bar{\psi}^{-}_{s1,d1} + \sum_{s2} e_2\bar{\psi}^{-}_{s2,d2}\right] = Q_2^{g'g} \tag{18}$$

where I_s is the adjacent region to boundary edge s. The first and the second term on the LHS stand for the sum of the angular flux moments in I_s and its upstream regions along track lines respectively in direction Ω_{d1} and Ω_{d2}. The third term is the incoming angular flux on edge s. The fourth and fifth terms are the incoming angular fluxes on edge $s1$ and $s2$, which are converted from the outgoing fluxes at the opposite endings on s. The RHS $Q_2^{g'g}$ is also a variation of the out-of-group source term. Actually, the coefficients e_i $(i = 1, 2)$ are the products of albedo factor, angular interpolation coefficients, and coefficient $e^k_{1,i}$ in Eq. (15); coefficients t_i $(i = 1, 2)$ are the products of albedo factor, angular interpolation coefficients, and coefficient $t^k_{\rho j,i}$ in Eq. (15).

So far, the MOC linear system, which is composed of Eqs. (17) and (18), has been established. It can be written in the form of operators as

$$A \cdot \begin{bmatrix} \vec{\phi} \\ \vec{\psi}^{ob,-} \end{bmatrix} = \vec{b} \tag{19}$$

where A is the coefficient matrix; $\vec{\phi}$ is the vector of angular flux moments; $\vec{\psi}^{ob,-}$ is the vector of incident angular fluxes on outer boundaries; and $\vec{b}$ is the RHS term.

2.1.3. Implementations

The coefficient matrix A of the linear system obtained above is only dependent on the geometric and material information; hence it can be computed before iterations, which saves time on recurrent sweeps. The RHS vector $\vec{b}$ is decomposed into the product of a matrix B and a new vector Q in our implementation, where B requires no update and can also be calculated in advance.

Matrix A is observed to be sparse and non-symmetric, and it is very time-consuming if solved by the conventional techniques like Gauss elimination or basic stationary iterative methods, especially when the matrix becomes larger. On the contrary, the Krylov subspace method could handle the problem efficiently and accurately. GMRES is selected from numerous methods of this kind because it performs better in numerical computation fields including neutron transport calculations. The basic idea of the algorithm is described in Fig. 2 (Saad, 1996). The GMRES solver is provided by Intel® Math Kernel Library. Furthermore, to take advantages of the sparsity in the matrix, the compressed sparse row (CSR) format storage scheme (Saad, 1996) is utilized in our code, which makes it less memory-intensive. The calculation flow of the Krylov subspace acceleration method is illustrated by Fig. 3.

2.2. The domain decomposition method

2.2.1. MOC linear system in sub-domains

In the domain decomposition scheme, the total calculation domain is partitioned into several sub-domains; and in each of them, a new linear system will be established and solved. The linear system in a sub-domain is slightly different from Eq. (19), because the type of boundaries is more complex. In a sub-domain, inner boundaries which are the common sides with the adjacent sub-domains exist. Both the incoming and outgoing angular fluxes have to be treated appropriately on the inner boundaries. In the equation system for a sub-domain, the incoming angular fluxes on inner boundaries are assumed to be known, and the outgoing ones will be solved from the linear system as unknowns. Thus the linear system in sub-domain *sub* can be represented as

$$A_{sub} \cdot \begin{bmatrix} \vec{\phi}_{sub} \\ \vec{\psi}^{ob,-}_{sub} \\ \vec{\psi}^{inb,+}_{sub} \end{bmatrix} = \vec{b}_{sub}\left(\vec{\psi}^{inb,-}_{sub}\right) \tag{20}$$

where the subscript *sub* is the sub-domain index; A is the matrix and $\vec{b}$ denotes the RHS vector which is a function of the incoming angular fluxes $\vec{\psi}^{inb,-}$ on inner boundaries; $\vec{\phi}$ and $\vec{\psi}^{ob,-}$ are moment vector and incoming flux vector on outer boundaries respectively; and $\vec{\psi}^{inb,+}$ indicates the vector of outgoing angular fluxes on inner boundaries.

1. *Compute* $r_0 = b - Ax_0$, $\beta := \|r_0\|_2$, *and* $v_1 := r_0/\beta$
2. *Define the* $(m+1) \times m$ *matrix* $\bar{H}_m = \{h_{ij}\}_{1 \le i \le m+1, 1 \le j \le m}$. *Set* $\bar{H}_m = 0$.
3. *For* $j = 1, 2, \ldots, m$ *Do:*
4. *Compute* $w_j := Av_j$
5. *For* $i = 1, \ldots, j$ *Do:*
6. $h_{ij} := (w_j, v_i)$
7. $w_j := w_j - h_{ij}v_i$
8. *EndDo*
9. $h_{j+1,j} = \|w_j\|_2$. *If* $h_{j+1,j} = 0$ *set* $m := j$ *and go to 12*
10. $v_{j+1} = w_j/h_{j+1,j}$
11. *EndDo*
12. *Compute* y_m *the minimizer of* $\|\beta e_1 - \bar{H}_m y\|_2$ *and* $x_m = x_0 + V_m y_m$.

Fig. 2. The GMRES algorithm.

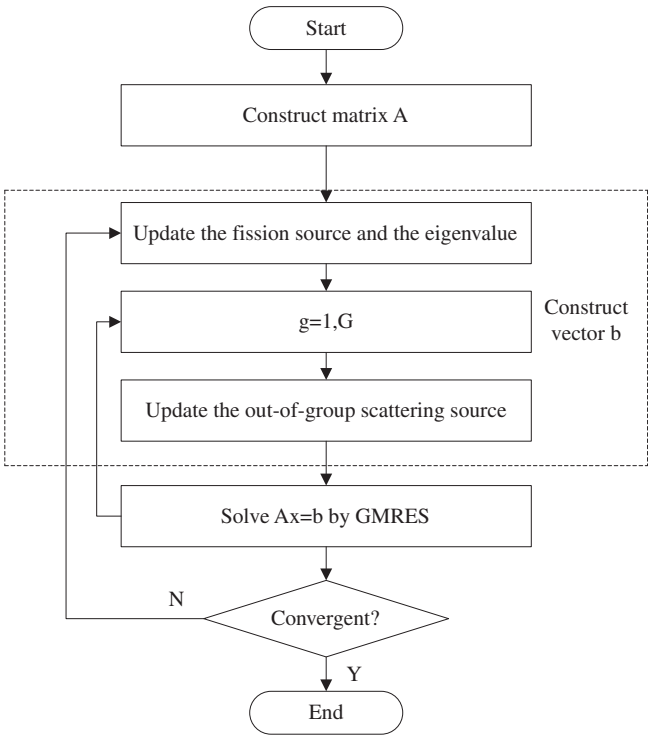

Fig. 3. Flow chart of Krylov subspace acceleration method.

The first two equations in the linear system above can be obtained from Eqs. (17) and (18) easily by distinguishing the incoming angular fluxes on inner boundaries from the ones on outer boundaries. Considering the former as known quantities and moving the relevant terms to the RHS, one can obtain

$$\left[a\phi_{\mathrm{P},I}+\sum_{\rho=0}^{2}\sum_{i'_k}\tau\phi_{\rho,i'_k}\right]+\sum_{k'_l}\varepsilon\cdot\psi^{ob,-}_{s,m}=\hat{Q}_1^{g'g}-\sum_{k''_l}\varepsilon\cdot\psi^{inb,-}_{k''_l,m} \tag{21}$$

and

$$\begin{aligned}&\left[\sum_{\rho=0}^{2}\sum_{i'_{d1}=1}^{I_s}t_1\phi_{\rho,i'_{d1}}+\sum_{\rho=0}^{2}\sum_{i'_{d2}=1}^{I_s}t_2\phi_{\rho,i'_{d2}}\right]\\&+\left[\psi^{ob,-}_{s,m}+\sum_{s1}e_1\cdot\psi^{ob,-}_{s1,d1}+\sum_{s2}e_2\cdot\psi^{ob,-}_{s2,d2}\right]\\&=\hat{Q}_2^{g'g}-\left[\sum_{k1}e_1\cdot\psi^{inb,-}_{k1,d1}+\sum_{k2}e_2\cdot\psi^{inb,-}_{k2,d2}\right]\end{aligned} \tag{22}$$

where $\psi^{inb,-}_{k1,d1}$ denotes the incoming angular flux from inner boundary along track line $k1$ in direction Ω_{d1}; the other symbols have the same meaning with the ones in Eqs. (17) and (18), although they are calculated only in a sub-domain instead of the total domain.

Eqs. (21) and (22) involve incoming angular fluxes from inner boundaries along track lines, which means the adjacent sub-domains are coupled line by line. Although this coupling method is precise, it will become burdensome when track lines are dense. An approximation is introduced to solve the problem. An inner boundary is subdivided into "inner-edges" by its adjacent regions on both sides. On each inner-edge, the angular flux is independent of space variables. As the scale of an inner-edge is no more than a flat-source region, the approximation is supposed to be reasonable. By introducing the approximation, the amount of coupling quantities declines. Moreover, various numerical experiments also demonstrate that the approximation will not bring an obvious effect on computational accuracy under common region partitions. In addition, the approximation permits the mismatches of track lines in adjacent sub-domains, which makes the ray tracing procedure more agile. According to the coupling method, the incoming angular fluxes from inner boundaries in Eqs. (21) and (22) can be classified by the inner-edges that the track lines' endpoints belong to.

The third equation in the linear system is for the outgoing angular fluxes on inner boundaries. It can be derived from Eq. (15) by converting the outgoing angular fluxes to the incoming ones at the opposite endpoint, which finally leads to

$$\psi^{inb,+}_{\hat{s}_1,m}+\sum_{k}\hat{e}_1\cdot\psi^{ob,-}_{s,m}+\sum_{\rho=0}^{2}\sum_{i_m}\hat{t}\phi_{\rho,i_m}=\sum_{k'}\hat{e}_2\cdot\psi^{inb,-}_{\hat{s}_2,m}+\hat{q}_i^{g'g} \tag{23}$$

where $\psi^{inb,+}_{\hat{s}_1,m}$ is the outgoing angular flux in direction m from inner-edge $\hat{s}_1$; k and k' denote the track lines, whose opposite endpoints belong to the outer boundary edge s and inner-edge $\hat{s}_2$ respectively; i is the adjacent region of inner-edge $\hat{s}_1$; i_m indicates region i 's upstream regions and i itself; $\hat{q}_i^{g'g}$ is a source term in region i.

Eqs. (21)–(23) are the components of Eq. (20), which is the linear system in a sub-domain. In the solving process, all the incoming angular fluxes on inner boundaries are assumed to be known as initial values. By solving Eq. (20) in sub-domains, outgoing quantities on inner boundaries are obtained and sent to the adjacent sub-domains as the incoming values for the next iteration step. The iteration will converge after several steps under a criterion on region fluxes and inner boundary fluxes.

2.2.2. Geometry processing and ray tracing

In AutoMOC, the geometry description and ray tracing are based on the customization of the graphic processing software AutoCAD, which enables a user to handle the complex geometry conveniently. In the scheme of domain decomposition for MOC, the innovation is inherited. Besides the existing functions, a new function on the geometry processing of domain decomposition is developed. One can finish the geometry description for MOC domain decomposition by following the steps below.

First, a user should draw a graph of the problem's geometry and partition it into regions in the interface of AutoCAD, which is a necessary step in both the old and new version geometry pretreatment tools. Then a little additional work has to be done by the user to describe the partition manner of sub-domains. In this step, one has only to specify regions to their corresponding sub-domains by some mouse box selections. In theory, the shape of a sub-domain is arbitrary. However, one has to consider the load balance in the subsequent parallelization. A simple solution is to ensure the areas of the sub-domains are equal approximately. Furthermore, automatic partition schemes are also available. After the partitions, different colors are painted in different sub-domains automatically for the convenience of distinction.

The last step is ray tracing, which can be completed by running a macro developed in the visual basic for application (VBA) language. Thanks to the inner-edge coupling scheme mentioned above, the ray tracing procedure can be carried out sub-domain by sub-domain, which is agile and convenient. The procedure finally produces a ray-tracking file for each sub-domain as the geometrical input.

2.2.3. Parallelization

The DD method has been proved to be a parallelizable algorithm by nature. In our study, the single program multiple data (SPMD) parallel scheme is adopted. It means that the computing tasks in sub-domains are assigned to several central processing units (CPU) averagely. All the CPUs are equal in status, and no host CPU exists. This scheme saves the communication with the host and is easy to implement.

Generally, once communication costs as much time as millions of floating-point operations. Hence, for a typical parallel cluster computer, communication is much more time-consuming than

computation. Therefore, the reduction of communication frequency is very necessary for improving the parallel efficiency. In the DD method, communication between adjacent sub-domains occurs at the end of each sub-domain iteration step. Thus the communication frequency depends on the number of sub-domain iteration steps. In the conventional scheme, sub-domain iteration is placed inside the energy group scanning calculation. It means that communication happens several times even in one of the energy groups in an outer iteration step. An adjustment is made in the computation flow: the sub-domain iteration loop is pulled out of the energy group sweeping, and the resulting flow is illustrated by Fig. 4. Consequently, the communication frequency is reduced, which is also confirmed by our numerical experiments. Actually, the phenomenon is observed in the experiments: it often converges most rapidly when the frequency of sub-domain iteration is limited to only once in each outer iteration step. The conclusion provides a default setting for the future computational schemes.

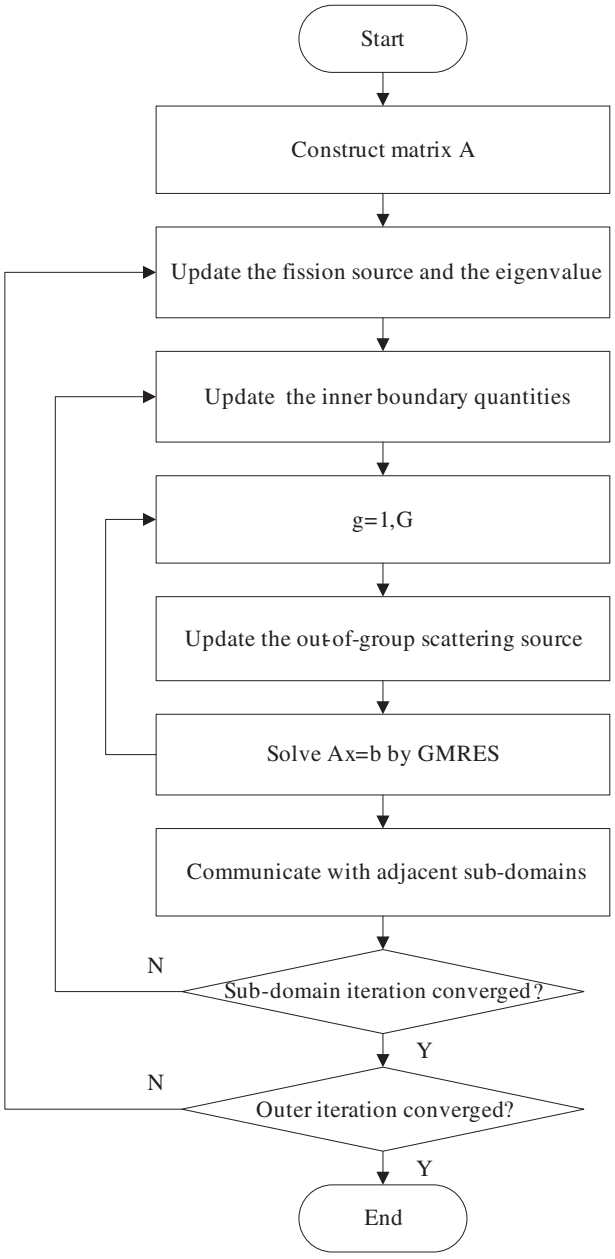

Fig. 4. Computation flow in a certain sub-domain.

The Message Passing Interface (MPI) parallel programming is adopted to achieve the above process.

3. Numerical results

Based on the aforementioned theoretical model, the 2D MOC program AutoMOC is updated to a new version. Hence three computational schemes are available: the MOC scheme, the accelerated MOC scheme, and the domain decomposition scheme based on accelerated MOC.

Four problems are computed by the schemes to evaluate the acceleration effect. First, the 69-group Pressurized Water Reactor (PWR) cell and assembly problems are computed respectively to exhibit the acceleration effect of Krylov subspace method in assembly-level problems. Second, a self-designed unstructured problem is proposed to prove the geometrical flexibility. Meanwhile, it is also a collision-dominated problem with a high scattering ratio. Then, the 2D C5G7 benchmark problem is calculated to demonstrate the applicability of the method for multi-assembly heterogeneous problem with favorable accuracy. Finally, a 1/4 300 MWe PWR core is considered as a validation for a core-level problem with full size.

All the numerical results are computed on the cluster with Xeon® E5405 2.00 GHz CPUs and 1 gigabit Ethernet connections.

3.1. The PWR cell and assembly problems

This is a 69-group problem. The PWR assembly includes 289 cells of the same type. Each cell is composed of three materials (Fig. 5). The partition scheme in a single cell is illustrated respectively by Fig. 6a for cell calculation and Fig. 6b for assembly calculation. In the cell calculation, 8 uniform azimuthal angles (0–π) and 2 optimal polar angles (0–$\pi/2$) (Leonard and McDaniel, 1995) are utilized. The width of a track line is 0.005 cm. The scheme is called 8-2-0.005 for convenience hereafter. For the assembly calculation, the 6-2-0.01 scheme is adopted. All the boundaries are reflective. The results are presented in Table 1.

Two schemes, MOC and accelerated MOC, are employed in the calculations respectively. As different discretized schemes are adopted, a discrepancy of the infinite multiplication factors exists between the cell calculations and the assembly ones. The results in Table 1 indicate that both the number of outer iterations and the computing time are reduced by the Krylov subspace acceleration method without loss of accuracy. Since no characteristics sweep exists in the iterative process of acceleration calculation, the reduction effect of computing time is greater than the one of the number of outer iterations. As the scale gets larger, the speedup decreases sharply in the assembly calculation. However, the saving of the absolute computing time is still significative in a 69-group assembly calculation. It demonstrates that the Krylov subspace acceleration is effective in cell and assembly calculations.

3.2. The unstructured geometry problem

This is a self-designed unstructured geometry problem with a vacuum boundary. It is intractable for some modular-ray-tracing MOC programs and their acceleration methods. Each of the shadowed regions in Fig. 7 is filled with a unique fuel material, and a region brimming with moderator surrounds them. The 2-group cross-sections are designed with high scattering ratios. The circle is sub-divided uniformly in the DD scheme. On the basis of the 8-2-0.01 scheme, the results are presented in Table 2.

The conventional MOC scheme converges very slowly in such a problem. It takes even more iterations than some core-level problems. However, the situation is ameliorated dramatically by the

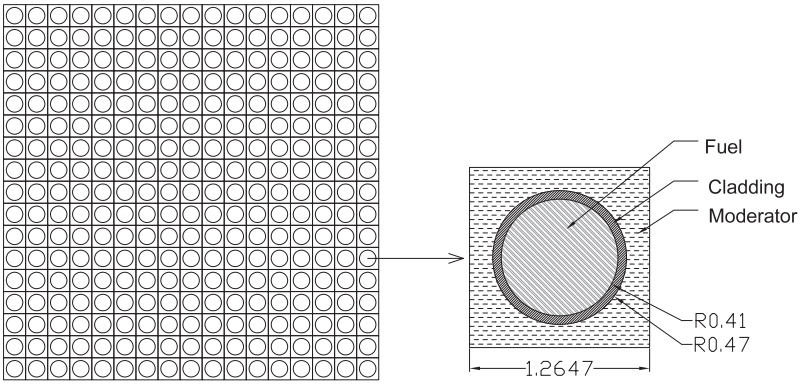

Fig. 5. Geometry of the 17 × 17 PWR assembly (cm).

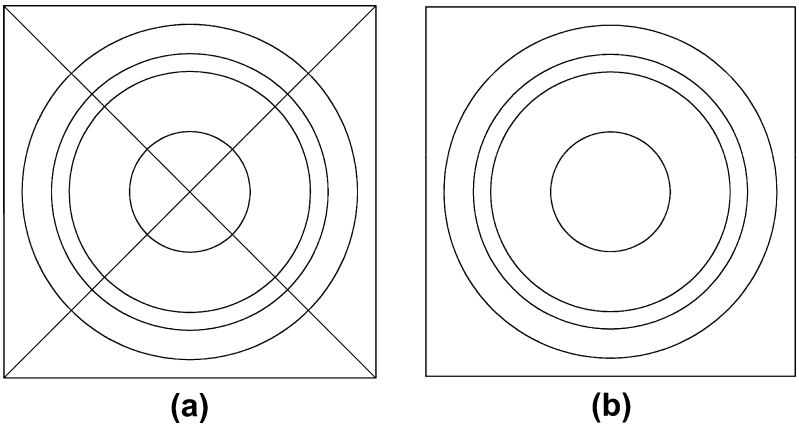

Fig. 6. Region partitions for the PWR cell and assembly problems.

Table 1
The results of the PWR cell and assembly problems.

Scheme	k_{inf}	Number of iterations	Computing time (s)	Speedup
MOC[a]	1.372250	67	23.41	–
MOC + Krylov[a]	1.372338	50	1.20	19.51
MOC[b]	1.373443	59	1447.25	–
MOC + Krylov[b]	1.373484	50	556.33	2.60

[a] Cell calculation.
[b] Assembly calculation.

Table 2
The results of the unstructured geometry problem.

Scheme	k_{eff}	Number of iterations	Computing time (s)	Speedup	Speedup per CPU core
MOC	1.013450	193	1633.15	–	–
MOC + Krylov	1.015255	46	94.58	17.27	–
DD-2[a]	1.015213	61	71.79	22.75	11.37
DD-4	1.015149	65	32.59	50.11	12.53
DD-8	1.014700	125	30.99	52.70	6.59

[a] Number of processor cores (i.e. number of sub-domains) utilized in the DD scheme.

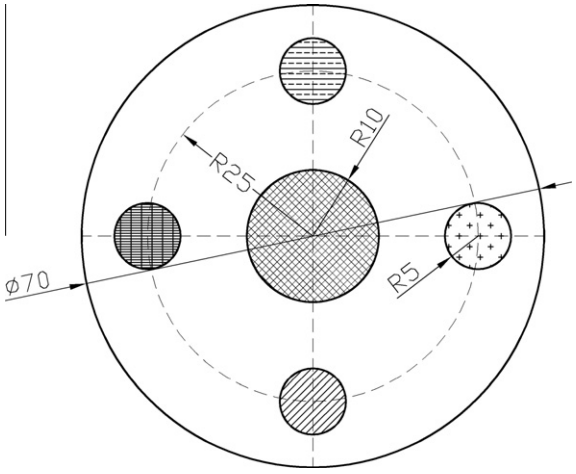

Fig. 7. Sketch of the unstructured geometry problem (cm).

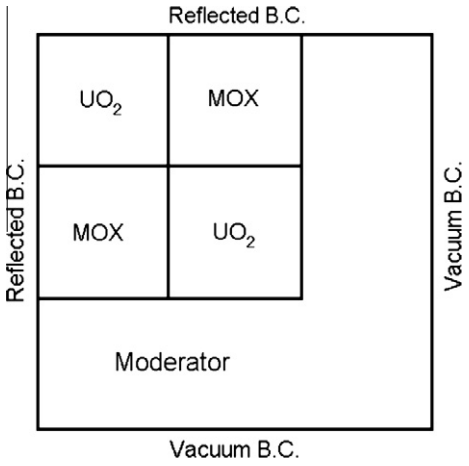

Fig. 8. Core configuration for the C5G7 benchmark problem.

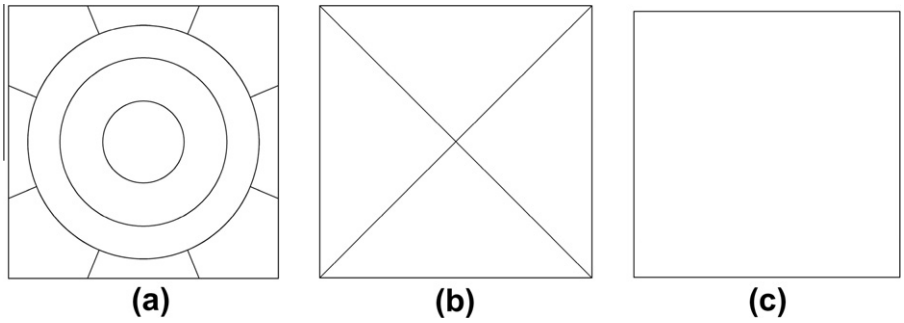

Fig. 9. Region partitions for the C5G7 benchmark problem.

Table 3
Efficiency comparisons for the C5G7 benchmark problem.

Scheme	Number of iterations	Computing time (s)	Speedup	Speedup per CPU core
MOC	74	4147.86	–	–
MOC + Krylov	54	–	–	–
DD-4	51	340.90	12.17	3.04
DD-9	81	203.45	20.39	2.27
DD-16	102	119.77	34.63	2.16
DD-25	125	90.92	45.62	1.82

Krylov subspace acceleration method. In addition, the DD method strengthens the acceleration effect. The quantity "speedup per CPU core" is introduced here to evaluate the accelerating contribution in a single processor core. It equals the result that the total speedup is divided by the number of processor cores occupied in the parallel computation. It is observed that both of the techniques are flexible in complex geometry and strong scattering problems.

3.3. The 2D C5G7 multi-assembly problem

This is the OECD/NEA 2D C5G7 benchmark problem (Fig. 8) (Lewis et al., 2001). The partition in a fuel-assembly cell is illustrated by Fig. 9a; each reflector cell nearby the fuel region is partitioned as Fig. 9b; the other reflector cells are not subdivided as shown in Fig. 9c. The scheme of 8-2-0.01 is adopted in the calculation.

In Table 3, the efficiencies of the schemes are presented respectively. As the Krylov subspace method fails completely in acceleration for the core-level problem, the computing time is not listed. However, the DD method based on the Krylov subspace acceleration performs quite well and gains favorable speedup. The speedup per CPU core decreases slowly as the number of processor cores increases. That is because the number of iterations rises as the subdomains becomes more. However, the total speedup is still in an uptrend.

In Table 4, the comparisons on accuracy are exhibited. All the relative errors are calculated based on the Monte Carlo reference. The errors are acceptable and comparable to the ones obtained by the other famous MOC programs (Lewis et al., 2001). Thus, the precision of all our three methods is quite reliable.

3.4. The 1/4 PWR core problem

This is a full-size 300 MWe PWR core problem referring to the Qinshan Nuclear Power Plant in China. The calculation is under the 1/4 core configuration in the first cycle as shown in Fig. 10. Three types of 15×15 assemblies with different fuel enrichments are loaded in the core. The figure labeled on an assembly indicates the poison arrangement scheme in it, which is illustrated in Fig. 11 in detail. Under the scheme of 6-2-0.01 in the 2-group problem, two types of pin-by-pin calculation are carried out. The first one is based on the homogenized cells under the partition in Fig. 9c. The second calculation is finished without spatial homogenization; the partitions in a fuel-assembly cell and a reflector cell follow Figs. 6b and 9c respectively. The results of the two cases are listed in Tables 5 and 6.

Because of the homogenization, the computations are less time-consuming in Table 5. Hence the Krylov subspace acceleration method is effective. Furthermore, the DD method gains excellent results both in total speedups and average speedups. The growth of the sub-domain amount causes the increase of iterations. Therefore, the speedup per CPU core has the optimal value, but the total speedup is still on the increase.

The core-level 2D heterogeneous calculation is still attractive as one of the trends in accurate analysis of reactor physics. In the heterogeneous case, the Krylov subspace method fails again in the acceleration for core-level problems, thus it is not mentioned in Table 6. Contrarily, the DD schemes overcome the difficulties and bring remarkable speedups. The optimal value of the speedup in a single processor core also exists.

Table 4
Accuracy comparisons for the C5G7 benchmark problem.

Scheme	Eigenvalue*		Specific pin power			Assembly power			Pin power distribution		
	k_{eff}	Error (%)	Max. pin power error (%)	Min. pin power error (%)	Max. error (%)	Inner UO_2 power error (%)	MOX power error (%)	Outer UO_2 power error (%)	AVG (%)	RMS (%)	MRE (%)
MOC	1.185363	−0.100	−0.27	0.63	3.84	−0.16	0.24	−0.17	0.47	0.72	0.38
MOC + Krylov	1.185459	−0.092	−0.19	0.36	3.73	−0.10	0.22	−0.29	0.46	0.70	0.37
DD-4	1.185464	−0.092	−0.20	0.31	3.70	−0.10	0.22	−0.31	0.47	0.71	0.37
DD-9	1.185471	−0.091	−0.22	0.35	3.78	−0.12	0.24	−0.28	0.47	0.71	0.38
DD-16	1.185452	−0.093	−0.24	0.31	3.72	−0.13	0.24	−0.28	0.47	0.71	0.38
DD-25	1.185471	−0.091	−0.26	0.32	4.13	−0.15	0.26	−0.26	0.47	0.72	0.38

* Reference k_{eff}: 1.186550.

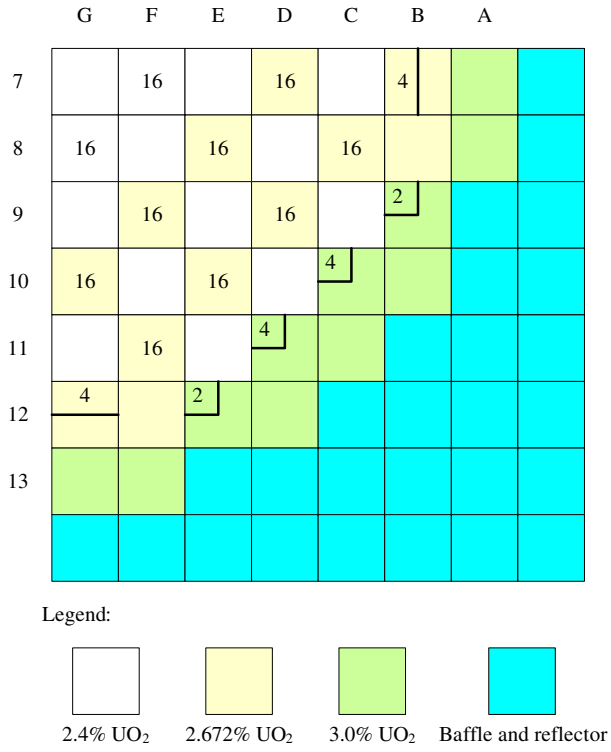

Fig. 10. The 1/4 core configuration in the first cycle of a 300 MWe PWR.

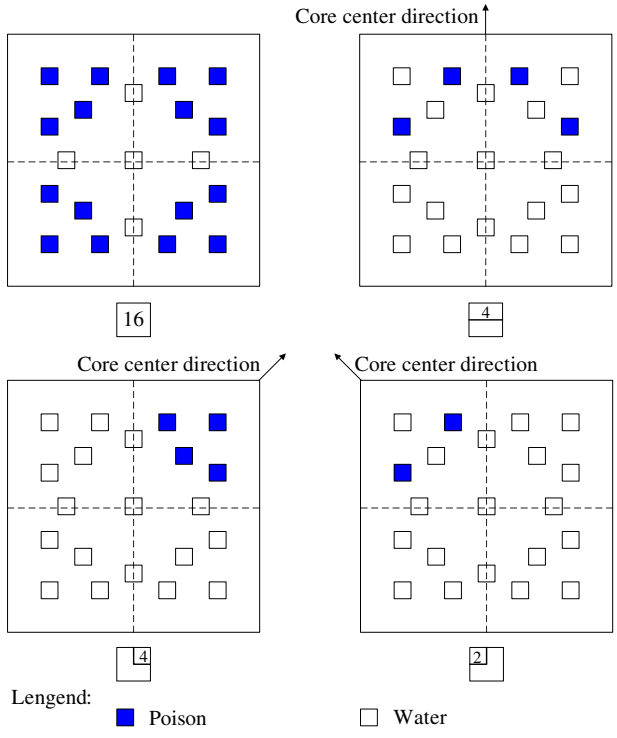

Fig. 11. The poison layout in 300 MWe PWR assemblies.

The total memory consumed in the calculations is also listed here. Since the storage of matrices is necessary, the acceleration methods cost more memory than the conventional MOC. However, the code becomes less memory-intensive as the sub-domain amount grows. This phenomenon can be attributed to the decoupling effect in the domain decomposition process. Taking a certain region for example, the situation of its directly-related regions is surely more complex when it is allocated in a larger sub-domain. Generally, it means that the number of non-zero elements in matrices decreases as the sub-domain amount increases. In this regard, the domain decomposition method improves the Krylov subspace acceleration again.

Table 5
The results of the 300 MWe PWR 1/4 core problem with cell homogenizations.

Scheme	k_{eff}	Number of iterations	Computing time (s)	Speedup	Speedup per CPU core	Memory consumed (MB)
MOC	1.168231	156	5368.38	–	–	231
MOC + Krylov	1.168348	86	1056.73	5.08	–	2876
DD-9	1.168300	139	68.78	78.05	8.67	791
DD-16	1.168271	169	34.43	155.92	9.75	679
DD-36	1.168223	205	20.25	265.11	7.36	624

Table 6
The results of the 300 MWe PWR 1/4 core problem without cell homogenization.

Scheme	k_{eff}	Number of iterations	Computing time (s)	Speedup	Speedup per CPU core	Memory consumed (MB)
MOC	1.165809	157	16528.1	–	–	954
DD-9	1.165934	209	900.13	18.36	2.04	3614
DD-16	1.165905	246	485.94	34.01	2.13	2712
DD-36	1.165858	313	288.33	57.32	1.59	1854

4. Conclusions

An acceleration technique based on Krylov subspace method and spatial domain decomposition method is proposed in this paper for the acceleration of arbitrary geometry MOC programs. In this technique, a MOC linear algebraic equation system is constructed to replace the conventional recurrent characteristics sweeping scheme. GMRES, which is a Krylov subspace method, is selected to solve the linear system efficiently. In order to overcome the defects of the Krylov subspace acceleration method in handling large-scale problems, non-overlapping spatial domain decomposition method is applied to reduce the scale of matrices. A proper technique is introduced to couple the adjacent sub-domains together. Parallel technology is utilized after a calculation flow adjustment and matched geometry processing tool is developed.

Four problems are computed and the results are analyzed. The Krylov subspace acceleration method is effective in multigroup assembly calculations, especially in collision-dominated cases. With the assistance of domain decomposition method, core-level computations can be executed efficiently and accurately. Moreover, favorable results are also obtained in unstructured geometry.

The phenomenon that the number of iterations increases as the sub-domain amount becomes more is observed. Some other advanced techniques on acceleration and parallelization may be applied to reduce the iterations and enhance the scalability.

Acknowledgments

The first author is grateful to Dr. Qichang Chen in Shanghai Nuclear Engineering Research and Design Institute (SNERDI) for his significant work on AutoMOC, which is the solid foundation of the following acceleration work.

This research was carried out under the financial support of the National Science Foundation of China (Approved number 10976021 and 10875094), the National High Technology Research and Development Program ('863' Program) of China (Approved

number 2009AA050705), and the National Magnetic Confinement Fusion Science Program (No. 2010GB111007).

References

Askew, J.R., 1972. A characteristics formulation of the neutron transport equation in complicated geometries. AEEW-M, 1108.

Chai, X., Yao, D., et al., 2010. Generalized coarse-mesh finite difference acceleration for the method of characteristics. In: Proceedings of the PHYSOR 2010, Paper 68_3 on CD-ROM, May 9–14, Pittsburgh, USA.

Chen, Q., Wu, H., Cao, L., 2008. Auto MOC – A 2D neutron transport code for arbitrary geometry based on the method of characteristics and customization of AutoCAD. Nucl. Eng. Des. 238, 2828–2833.

Cho, J.Y., Kim, K.S., Shim, H.J., et al., 2008. Whole core transport calculation employing hexagonal modular ray tracing and CMFD formulation. J. Nucl. Sci. Technol. 45, 740–751.

Dahmani, M., Tellier, R.L., et al., 2005. An efficient preconditioning technique using Krylov subspace methods for 3D characteristics solvers. Ann. Nucl. Energy 32, 876–896.

Grassi, G., 2007. A nonlinear space-angle multigrid acceleration for the method of characteristics in unstructured meshes. Nucl. Sci. Eng. 155, 208–222.

Halsall, M.J., 1980. CACTUS, a characteristics solutions to the neutron transport equations in complicated geometries. AEEW-R-1291, UK Atomic Energy Authority.

Halsall, M.J., 1998. WIMS8 – speed with accuracy. ANS Topical Meeting, October 1998, Long Island, USA.

Joo, H.G., Cho, J.Y., et al., 2002. Dynamic implementation of the equivalence theory in the heterogeneous whole core transport calculation. In: Proceedings of the PHYSOR 2002, Paper 13A-02 on CD-ROM, October 7–10, Seoul, Korea.

Knott, D., Forssen, B.H., et al., 1995. CASMO-4 Methodology Manual. STUDSVIK/SOA-95/02, Studsvik of America.

Kosaka, S., Saji, E., 2000. Transport theory calculation for heterogeneous multi-assembly problem by characteristics method with direct path linking technique. J. Nucl. Sci. Technol. 37, 1015–1023.

Lee, G.S., Cho, N.Z., et al., 2000. Acceleration and parallelization of the method of characteristics for lattice and whole-core heterogeneous calculations. In: Proceedings of the PHYSOR 2000, Paper 091 on CD-ROM, May 7–12, Pittsburgh, USA.

Leonard, A., McDaniel, C.T., 1995. Optimal polar angles and weights. Trans. Am. Nucl. Soc. 73, 172–173.

Lewis, E.E., Smith, M.A., et al., 2001. Benchmark Specification for Deterministic 2-D/3-D MOX Fuel Assembly Transport Calculations Without Spatial Homogenisation (C5G7 MOX). OECD/NEA Report, NEA/NSC/DOC (2001)4.

Saad, Y., 1996. Iterative Methods for Sparse Systems. PWS Publishing Company, Boston, USA.

Sanchez, R., Mondot, J., et al., 1988. APOLLO2: a user-oriented, portable, modular code for multigroup transport assembly calculations. Nucl. Sci. Eng. 100, 352–362.

Santandrea, S., Sanchez, R., 2002. Acceleration techniques for the characteristic method in unstructured meshes. Ann. Nucl. Energy 29, 323–352.

Tang, C., Zhang, S., 2009. Development and verification of an MOC code employing assembly modular ray tracing and efficient acceleration techniques. Ann. Nucl. Energy 36, 1013–1020.

Yamaji, K., Matsumoto, H., et al., 2010. Simple and efficient parallelization method for MOC calculation. J. Nucl. Sci. Technol. 47, 90–102.

Yamamoto, A., 2005. Generalized coarse-mesh rebalance method for acceleration of neutron transport calculations. Nucl. Sci. Eng. 151, 274–282.