Annals of Nuclear Energy 99 (2017) 427–433

Contents lists available at ScienceDirect

Annals of Nuclear Energy

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

Development and verification of MOC code based on Krylov subspace and efficient preconditioning techniques

Yong Zheng, Minjun Peng *, Genglei Xia

Fundamental Science on Nuclear Safety and Simulation Technology Laboratory, Harbin Engineering University, Harbin 150001, China

ARTICLE INFO

Article history:
Received 18 January 2016
Received in revised form 30 April 2016
Accepted 28 September 2016
Available online 20 October 2016

Keywords:
Method of characteristics
Krylov subspace
Preconditioned GMRES(m)

ABSTRACT

Derived from the conventional characteristic sweeping operations, the method of characteristics (MOC) with matrix form has more favorable performance compared with its traditional implementation potentially. However, the advantage depends heavily upon the efficiency of the linear algebraic solver. In current study, a simple and efficient preconditioning technique is implemented in the restart version of Generalized Minimal RESidual algorithm (GMRES) to accelerate the resulted linear system. A complete code including geometry processing and algebraic solver has been developed and verified with the reference benchmark problem. Numerical results demonstrate the code can model the reference problem accurately, and the proposed preconditioning techniques based on the typical iterative method can decrease dramatically the number of iterations without introducing additional computation and storage expense.
© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Since the method of characteristic (MOC) has been proposed (Askew, 1972), there have been a mount of researches for it due to its geometrical flexibility and natural parallel potential. At present, both the latest developed codes and the upgrade versions of previous softwares have introduced the MOC computation module to handle the arbitrary geometry for two dimension or three dimension case, such as DeCART (Joo et al., 2004), CRX (Hong and Cho, 1998), Chaplet (Kosaka and Saji, 2000), DRAGON (Marleau et al., 2011), CASMO-4 (Smith and Rhodes, 2002). However, the acceptable results can only be obtained by sweeping the neutron travel tracks once and again until the outer iteration reaches convergence. To reduce the number of outer iterations, the method of coarse mesh finite difference (CMFD) has been adopted in great majority of transport codes, and numerical experiments have demonstrated CMFD is a computationally inexpensive and efficient acceleration technique. Nevertheless, the time-consuming sweeping operations seemingly indicate conventional MOC is still not a efficient method. Derived from the recurrent sweeping operations, the linear algebra form of MOC can improve computation efficiency essentially (Zhang et al., 2011), because it permits one to take advantage of more acceleration techniques in linear algebra field which can provide the additional latest and efficient numerical algorithm to solving transport equation. Therefore the focus problems are transformed into how to establish the linear algebra system (including precalculation coefficient matrices/right hand sides) and solve the linear system efficiently.

Although the configurations of coefficient matrices are irregular for transport problems generally, one could still find some regularities particularly for the application arisen from MOC, which will reduce the time establishing the coefficient matrices. Furthermore, the coefficient matrices even become the consequents after applying elementary transformations to symmetric matrices if the neutron travel tracks can be mapped with special pattern (Wu et al., 2014). To solve the final large sparse linear system, the Generalized Minimal RESidual (GMRES) algorithm is a popular choice among iterative methods (Takeda and Kitada, 2012). Details about this algorithm can be found in the original paper (Saad and Shultz, 1986). As a variety of Krylov subspace method, GMRES is an attractive technique in sparse matrix computation field since being proposed because it can solve unsymmetric linear system elegantly. However, it is observed that is not efficient enough when the coefficient matrices become larger for the current transport issue. It is necessary to propose a acceleration approach to improve its convergence behavior. From the numerical perspective, the principal topic is to find an appropriate preconditioner which can transform the original linear system into one with the same solution, but that is likely to be easier to solve with the same iterative solver.

When matrix is less sparse, choosing an appropriate preconditioner can be difficult. In particular, it is of vital importance to keep the sparsity and decrease computation cost for large sparse linear system. Arising from the discretization of PDEs of elliptic type, the

* Corresponding authors. Tel.: +86 45 182569611; fax: +86 45 182569622.
E-mail address: heupmj@163.com (M. Peng).

http://dx.doi.org/10.1016/j.anucene.2016.09.049
0306-4549/© 2016 Elsevier Ltd. All rights reserved.

incomplete LU factorization technique was developed originally for M-Matrix and become the most common preconditioner at present. Although elements filling within can be controlled by ILU(p) or ILU(τ, p), the memory requirement and computation time are prohibitively expensive with respect to large scale matrix.

In current work, a new MOC code employing preconditioned Krylov subspace method has been developed and verified against the reference benchmark, i.e. OECD/NEA C5G7 MOX (OECD, 2003). The rest of this paper is organized as follows: Section 2 exhibits the methodology of this study, including the overview of the code and the details about the acceleration technique, Section 3 displays the numerical experiments for the verification and evaluation of the implemented acceleration scheme, Section 4 makes the conclusions for the present work.

2. Theoretical model

The code can be divided into several modules: the first one is the geometry processing consisted of geometry representation and characteristic line tracing; the second is to establish the structure of coefficient matrices and insert nonzero elements by sweeping all neutron travel tracks; the last one is to solve the sparse linear system using preconditioned Krylov subspace method. All above modules were programmed based on C language.

2.1. Geometry processing

The information of tracks is obtained by a independent geometry processing module similar to OpenMOC (Boyd et al., 2014) with improvement which can efficiently determine the identifier of flat source region (FSR) that a given local coordinate resides within. Entire geometry is represented by a list of constructive solid geometry formulations (CSG) by which arbitrary complicated geometry can be constituted theoretically. The most elementary unit is the FSR (namely basic cell in code), which consists of one material identifier and several weighted curves named surfaces. Each weighted curve has only two weight values, i.e. −1 or +1, which separate the whole space into two half-spaces exactly, namely negative space or positive space, determined by the value of curve function, thus the curve becomes the interface of two FSRs. Entire geometry is constructed in a hierarchical fashion, including following nested entities: Cell, Universe, Lattice.

In order to avoid generality loss, the tracks need sweeping entire computational domain to generate the information of all segments. Contrasting with AMRT (Sanchez et al., 2002), this method needs more computation time and memory requirement, but it is absolutely necessary and worthwhile to handle arbitrary complicated geometry. Meanwhile this additional cost is negligible comparing with the iteration process of linear algebra system, and there is a typical remedy benefiting from the symmetry, thus only the tracks whose azimuths reside within in range (0–π) need to be considered, the remaining tracks have the identical information except for inverse direction. In OpenMOC, the FSRs' identifiers are searched in a hash table based on the alphabetical string of the FSR's hierarchical level. In current study, the hash table is eliminated and the FSR's identifier is determined in advance only based on the FSR's hierarchical level.

2.2. Matrix characteristics method

There are a mount of articles about the derivation of matrix MOC, the general concept is that outgoing angular flux can be expressed in term of incident angular flux and neutron source of the FSR. Continuously, the angular flux located on the boundary can be expressed in terms of incident flux and the neutron sources of all FSRs spanned by this track (the energy group index has been omitted for simplicity hereafter):

$$\psi_{l,k}^{out} = \psi_k^{in} e_{0,l} + \sum_{j=0}^{l} \frac{Q_j}{\Sigma_{t,j}} \left[1 - exp\left(-\Sigma_{t,j} s_j\right)\right] e_{j+1,l} \tag{1}$$

where ψ_k^{in} and $\psi_{l,k}^{out}$ denote the incident and outgoing angular flux of track k on the outer boundary respectively, and I is the number of FSRs spanned by this track; Q_j represents the neutron source in FSR j which has been spanned by track k; $e_{i,j}$ represents the exponential attenuation coefficient from region i to j along the track k.

$$\begin{cases} Q_j^g = \sum_{g'=1}^{G} \Sigma_{s,g'g} \phi_{g'} + \frac{\chi_g}{k_{eff}} \sum_{g'}^{G} \nu \Sigma_{f,g'} \phi_{g'} \\ e_{i,j} = \exp\left(-\Sigma_{t,i} s_i - \Sigma_{t,i+1} s_{i+1} \ldots - \Sigma_{t,j} s_j\right) \end{cases} \tag{2}$$

The track's average angular flux in any FSR can be expressed in terms of its incident flux and the FSR's source, furthermore the FSR's average angular flux can be calculated in terms of the "width" of segments and their average angular flux. After taking weighted sum of whole average angular flux located in FSR i, one can obtain the scalar flux of FSR i:

$$\phi_i = f_1(Q_i) + \sum_k \left[f_2(\psi_k) + \sum_j f_3(Q_j) \right] \tag{3}$$

Specifically, after distinguishing the ingroup scattering term from total source in FSR i and rearranging the Eq. (3), the following matrix equation can be obtained for whole FSRs:

$$(S_1 + D_1)\Phi + S_1' \Psi = S_1 Q \tag{4}$$

where S_1 denotes the sweeping matrix by $n \times n$ for scalar flux contribution between FSRs and S_1' represents the sweeping matrix by $n \times m$ for angular flux contribution; D_1 is a diagonal matrix by $n \times n$ whose elements are the differences between transport and self-scattering cross sections; Φ, Ψ, Q are the vectors of scalar flux, angular flux and the outgroup source respectively, the n and m are the number of FSRs and tracks respectively.

It should be pointed out that only those FSRs connected by identical track can make contribution to each other, so the matrix S_1 and S_1' would definitely be sparse, the sparsity depends upon discretization scheme for direction, domain and the density of the tracks. Consequently, it is difficult to determine the number and positions of nonzero elements in compress sparse row (CSR) storage format prior to sweeping the tracks. The two-step strategy is adopted in current study: the matrix structure is constituted firstly by pre-sweeping tracks' information, then the coefficients are calculated and inserted in the proper positions defined by the matrix structure. Thanks to that matrix structure is shared by all energy groups, it can reduce memory and computation expense lightly.

The additional m equations to close the algebra system are provided through implementing the outer boundary condition to Eq. (1), which could result in Eq. (5):

$$S_2 \Phi + (E + S_2') \Psi = S_2 Q \tag{5}$$

where S_2, S_2' denote the contributions of scalar flux and incident angular flux to reflection tracks' incident angular flux respectively.

It should be noted that $S_1 = S_1^T D$, $S_2 = P S_1^T$ where D denotes the $n \times n$ diagonal matrix whose elements are determined by the FSRs' scattering ratios, thus what needs to be calculated is only the upper triangular part of the S_1 when generating matrix for each group, and the P denotes the permutation matrix which swaps the rows appropriately and should result in the identity matrix E, S_2' has only one element in each row which denotes the contribution of track's incident angular flux to outgoing flux (i.e. reflection track's

Download English Version:

https://daneshyari.com/en/article/8067436

Download Persian Version:

https://daneshyari.com/article/8067436

Daneshyari.com