Implementation of Preconditioned Krylov Subspace Method in MATRA Code for Whole Core Analysis of SMART

Hyuk Kwon [a*] , S. J. Kim, J. P. Park, and D. H. Hwang
[a] *SMART Development Division, Korea Atomic Energy Research Institute*
, 150, Dukjin-dong, Yuseong, Daejeon, 305-353, Korea
[*] *Corresponding author: kwonhk@kaeri.re.kr*

1. Introduction

Krylov subspace method was implemented to perform the efficient whole core calculation of SMART with pin by pin subchannel model without lumping channel. The SMART core consisted of 57 fuel assemblies of 17 by 17 arrays with 264 fuel rods and 25 guide tubes and there are total 15,048 fuel rods and 16,780 subchannels.

Restarted GMRES and BiCGStab methods are selected among Krylov subspace methods. For the purpose of verifying the implementation of Krylov method, whole core problem is considered under the normal operating condition. In this problem, solving a linear system $Ax = b$ is considered when A is nearly symmetric and when the system is preconditioned with incomplete LU factorization(ILU). The preconditioner using incomplete LU factorization are among the most effective preconditioners for solving general large, sparse linear systems arising from practical engineering problem.

The Krylov subspace method is expected to improve the calculation effectiveness of MATRA code rather than direct method and stationary iteration method such as Gauss elimination and SOR. The present study describes the implementation of Krylov subspace methods with ILU into MATRA code.

2. Methods and Results

2.1 Preconditioned Krylov Subspace Methods

Original numerical scheme in MATRA code is Successive Over-Relaxation(SOR) with fixed under-relaxation factor. Acceleration and robustness for converging solution is used with preconditioned GMRES and BiCGStab[1]. The sparse matrix generated by momentum and energy conservation equations are stored in compressed sparse row(CSR) format. Some linear algebra such as matrix-vector product, inner-products is operated in the CSR format.

In the field of computational fluid dynamics, these two iterative methods, BiCGStab and GMRES are most promising among the Krylov subspace methods and are representative. The currently de facto standard for unsymmetric system is the GMRES method. In this method x_i , in the Krylov subspace of dimension i , is constructed for which the norm of residual is minimal based on the Arnoldi algorithm. The price to pay for unsymmetry is that one has to store a full orthogonal basis for the Krylov subspace, which means the more iterations are done the more basis vectors have to be stored. For these reasons, full GMRES is unfeasible. One remedy for memory disadvantage restricts the work per iteration. As shown in Fig. 1, restarted GMRES with preconditioner K , GMRES(m), is incorporated in the MATRA code.

$$r = b - Ax^{(0)}, \text{for a given initial guess } x^{(0)}$$
$$\textbf{Do } j = 1, 2, \ldots$$
$$\beta = \|r\|_2,\ v^1 = r/\beta,\ \hat{b} = \beta e^1$$
$$\textbf{Do } i = 1, 2, \ldots, m$$
$$z_i = K_i^{-1}\tilde{v}^i;\ w = Az_i$$
$$\textbf{Do } k = 1, 2, \ldots, i$$
$$h_{k,i} = \left(v^k\right)w\ ; w = w - h_{k,i}v^k$$
$$h_{i+1,i} = \|w\|_2;\ v^{i+1} = w / h_{i+1,i}$$
$$\textit{old Givens rot's on new column:}$$
$$\textbf{Do } k = 2, \ldots, i$$
$$\gamma = h_{k-1,i}$$
$$h_{k-1,i} = c_{k-1}\gamma + s_{k-1}h_{k,i}$$
$$h_{k,i} = -s_{k-1}\gamma + c_{k-1}h_{k,i}$$
$$\textit{new Givens rot's for removing } h_{i+1,i}$$
$$\delta = \sqrt{h_{i,i}^2 + h_{i+1,i}^2}, c_i = h_{i,i}/\delta,\ s_i = h_{i+1,i}/\delta$$
$$h_{i,i} = c_i h_{i,i} + s_i h_{i+1,i}$$
$$\hat{b}_{i+1} = -s_i\hat{b}_i,\ \hat{b}_i = c_i\hat{b}_i$$
$$\rho = \left|\hat{b}_{i+1}\right|$$
$$\textbf{if } \rho \textit{ is small enough } \textbf{then}$$
$$,\mathrm{n}$$
$$\textbf{enddo } i$$
$$\textit{Solution of } H_{n,n}y = \hat{b}$$
$$n = m,\ y_n = \hat{b}_n / h_{n,n}$$
$$SOL: \textbf{Do } k = n-1, \ldots, 1$$
$$y_k = \left(\hat{b}_k - \sum_{i=k+1}^{n} h_{k,i}y_i\right) / h_{k,k}$$
$$x = \left(\sum_{i=1}^{n} y_i v^i\right) \text{,if } \rho \text{ small enough quit}$$
$$r = b - Ax$$

Fig.1. Preconditioned GMRES(m) algorithm[1]

The generalization of conjugate gradients for unsymmetric system, Bi-CG, displays often a quite irregular convergence behavior. Sonneveld recognized that the transpose A matrix operation could be used for a further reduction of the residual, by a minor modification of Bi-CG scheme. This method is known as the CGS. In 1992, van der Vorst showed that Bi-CG could be combined minimal residual steps such as GMRES. This resulted in the Bi-CGStab and this method is illustrated as shown in Fig. 2 with preconditioner K.

Compute $r^{(0)} = b - Ax^{(0)}$ for some initial guess $x^{(0)}$
Choose $\hat{r}^{(0)} = r^{(0)}$
Do $i = 1, 2, \ldots$
  $\rho_{i-1} = \hat{r}^{(0)*} r^{(i-1)}$
  if $\rho_{i-1} = 0$ method fails
  if $i = 1$
    $p^{(i)} = r^{(i-1)}$
  else
    $\beta_{i-1} = (\rho_{i-1} / \rho_{i-2})(\alpha_{i-1} / \omega_{i-1})$
    $p^{(i)} = r^{(i-1)} + \beta_{i-1}\left(p^{(i-1)} - \omega_{i-1} v^{(i-1)}\right)$
  endif
  solve $\hat{p}$ from $K\hat{p} = p^{(i)}$
  $v^{(i)} = A\hat{p}$
  $\alpha_i = \rho_{i-1} / \left(\hat{r}^{(0)*}\right) v^{(i)}$
  $s = r^{(i-1)} - \alpha_i v^{(i)}$
  if $\|s\|$ small enough then
    $x^{(i)} = x^{(i-1)} + \alpha_i \hat{p}$ and stop
  solve z from $Kz = s$
  $t = Az$
  $\omega_i = s^* t / t^* t$
  $x^{(i)} = x^{(i-1)} + \alpha_i \hat{p} + \omega_i z$
  if $x^{(i)}$ is accurate enough then quit
  $r^{(i)} = s - \omega_i t$ if $\omega_i \neq 0$
enddo i

Fig.2. Preconditioned Bi-CGSTab algorithm[1]

The construction of effective and efficient preconditioners is largely problem dependent. Many different preconditioners have been suggested over the years, among all these preconditioner the incomplete LU factorizations[1] are the most popular ones. The basic idea of ILU is to modify Gaussian elimination to allow fill-ins at only a restricted set of positions in the LU factors. Basic ILU(0) is implemented into MATRA code as shown in Fig. 3.

ILU for an n by n matrix A
$S = \{(i, j) | a_{i,j} \neq 0\}$
Do $k = 1, 2, \ldots, n-1$
  $d = 1 / a_{k,k}$
  Do $i = k+1, k+2, \ldots, n$
    if $(i, k) \in S$
      $e = da_{i,k}$; $a_{i,k} = e$
      Do $j = k+1, \ldots, n$
        if $(i, j) \in S$ and $(k, j) \in S$
          $a_{i,j} = a_{i,j} - ea_{k,j}$
        endif
      end Do j
    endif
  end Do i
end Do k

Fig.3.ILU Preconditioner for a general matrix A[1]

2.2 Verification Problem

Verification of the implementation of Krylov subspace method in MATRA code is performed on the SMART-1/8 core and whole core. Verification cases to estimate performance of present method are as shown on Table 1.

Table 1: Verification problem case definition

CASE	channel	Axial node	Operating Condition
SMT-1/8 core	2331	50	Normal operating condition
SMT-WC	16780	50	Normal operating condition

Sparse pattern of present problem is a symmetric matix and with condition number of 270.15 it is well-conditioned problem. Sparsity of the SMT-WC problem is shown in Fig. 4.

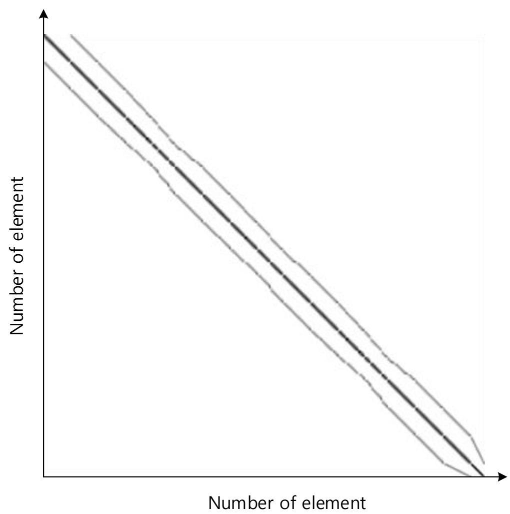


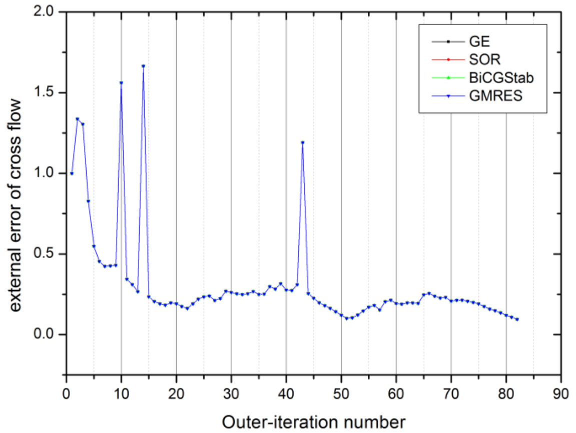
Fig.4. Sparsity of matrix on SMART whole core problem

2.3 Performance of Krylov Subspace Methods

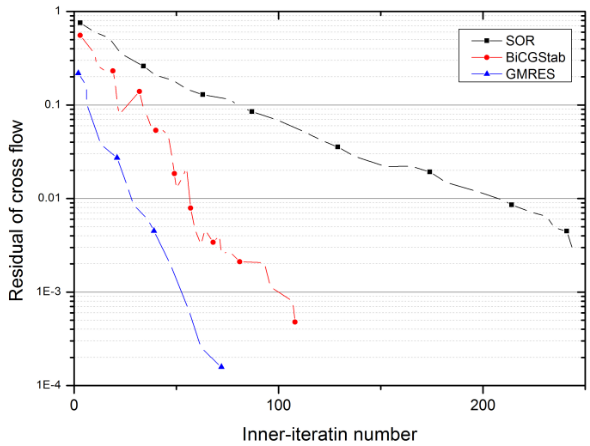
The performance of Krylov subspace methods was tested on the reference method SOR and direct solver of GE. Computation time is compared among different numerical linear solver such as SOR, GMRES with ILU, and BiCGStab with ILU as shown in Table 2.

Table 2: Performance results of linear solver on SMART application test

Case		SOR	GMRES(10)	BiCGStab
SMT-1/8 core	Inner-iteration (Outer)	102581 (82)	5721 (82)	7134 (82)
	Total-CPU(sec)	619	228.5	198
SMT-WC	Inner-iteration (Outer)	62825 (35)	34551 (35)	43085 (35)
	Total-CPU(sec)	2012	720	624



a) Outer-iteration history



b) Inner-iteration history

Fig.5. Iteration history of SMT-1/8 core problem with varying linear solver

SOR algorithm used the optimal over relaxation factor with fixed value of 1.6. BiCGStab method is highly efficient method in the view of CPU time. In Table 2, GMRES method performed the calculation with the least inner iteration however the calculation time is more consumed than that of BiCGStab due to requiring more Matrix-vector multiplication.

Iteration history of inner-iteration and outer-iteration for the problem of SMT-1/8 core is shown in Fig. 5. A solution of governing equation sets is obtained by the marching solution that is successively solved by plane by plane. In this algorithm, pressure and axial mass flow rates is updated on the outer-iteration and enthalpy and cross flow is calculated in inner-iteration. Considering the numerical scheme, Outer-iteration is not sensitive according to the various linear solvers.

In order to estimate the performance of linear solver, convergence speed for inner-iteration number is investigated at the plane of axial node 30 and outer-iteration 20. Inner-iteration number is strongly dependent on the linear solver as shown in Fig. 5(b). GMRES method is most efficient method aspects of convergence speed per iteration number. In the given maximum iteration number, SOR does not reached at the convergence criteria.

3. Conclusions

In this paper, we explore an improved performance of MATRA code for the SMART whole core problems by of Krylov subspace method. For this purpose, two preconditioned Krylov subspace methods, GMRES(10) and BiCGStab, are implemented into the subchannel code MATRA. A typical ILU method is used as the preconditioner. Numerical problems examined in this study indicate that the Krylov subspace method shows the outstanding improvements in the calculation speed and easy convergence.

REFERENCES

[1] Y. Saad, Iterative methods for sparse linear systems, SIAM, 2nd edition, 2003.