Investigating the Suitability of Preconditioned GMRES Method on GPU

2010008 Kenichi Ito

Numerical simulation involves solving **large and sparse simultaneous linear equations** obtained by discretizing partial differential equations describing physical phenomena using the finite difference method or the finite element method. Attempts are being made to perform numerical simulations in an on-site environment by utilizing a **GPU**, which are small in scale but have high computing performance, rather than large-scale computers such as clusters or supercomputers. For example, in the medical field, from the viewpoint of information management and simulation control, it may be desirable to use the computer of the own organization rather than the computer of the external organization.

GPUs are used as accelerators, but since the characteristics of CPUs and GPUs are different, it is expected that the suitable algorithm will also be different. Therefore, this research evaluated the preconditioned GMRES method on a GPU (NVIDIA A100 PCIe) and CPUs (HPC System "KAGAYAKI") to reveal GPU-friendly numerical solution methods for large and sparse simultaneous linear equations, obtained by discretizing differential equations with finite difference methods or finite element methods.

The iterative method, which is for solving simultaneous linear equations, starts from an appropriately chosen initial value and successively creates a sequence of approximate solutions that converge to the true solution. As an iterative method, there is a **Generalized Minimal RESidual (GMRES) method** based on the **Krylov subspace**. The GMRES method starts from the right-hand vector $\boldsymbol{b}$ and generates an orthonormal basis $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{j+1}$, expanding the Krylov subspace. It updates the approximate solution $\boldsymbol{x}_j$ so that the residual norm $\|\boldsymbol{r}_j\|_2 = \|\boldsymbol{b} - A\boldsymbol{x}_j\|_2$ is minimized at each step.

The convergence of the Krylov subspace method generally depends on the eigenvalue distribution of the coefficient matrix, and the smaller the eigenvalue distribution and the closer it is to 1 (identity matrix), the faster the convergence. To improve convergence and stability, the coefficient matrix is **ordered** and **preconditioned** before starting the iteration.

I compared the **Classical Gram-Schmidt 2 (CGS2)** and **Modified Gram-Schmidt (MGS)** methods, for orthogonalization used during iterations in the GMRES method on the GPU. The results of the evaluation by CUDA show that the CGS2 method on the GPUs has much shorter processing time than the MGS method with no convergence issues.

The **Incomplete LU decomposition (ILU)** preconditioned GMRES method was evaluated for ILU(0), ILU(1), or ILU(2), and it was found that

heavy preconditioning (such as ILU(2)) is required for ill-conditioned problems. Regarding the level at which fill-in is allowed in ILU, it was found that the appropriate level at which the processing time is minimized depends on the increase in the number of nonzero elements and convergence.

For ordering, **Reverse Cuthill-McKee (RCM)** and **Nested Dissection (ND)** were evaluated by the GMRES method. Although it depends on the matrix to be solved, in general, the application of RCM on CPUs improved convergence and reduced processing time, while the application of RCM on GPUs improved convergence but increased processing time. In the case of GPUs, it was found that using ND for ordering reduced the GMRES processing time in general, although it depends on the matrix to be solved.

前処理付き GMRES 法の GPU に対する適合性調査

2010008 伊藤 健一

数値シミュレーションでは、物理現象を記述する偏微分方程式を有限差分法や有限要素法などで離散化することで得られる**大規模かつ疎な連立一次方程式** $A\boldsymbol{x} = \boldsymbol{b}$ を求解することになる。クラスタやスーパーコンピュータといった大規模な計算機ではなく、小規模ながら高い演算性能を持つ **GPU** を活用して、オンサイト環境での数値シミュレーションを行う試みが行われている。例えば、医療分野では情報管理やシミュレーションのコントロールの点から外部組織の計算機よりも自組織の計算機での計算が望まれる場合がある。

アクセラレータとして GPU を用いるが、CPU と GPU では特性が異なるため、適したアルゴリズムも異なることが予想される。そのため、本研究では大規模かつ疎な連立一次方程式の GPU に適した数値解法を明らかにすべく、GPU (NVIDIA A100 PCIe) と CPU (HPC System “KAGAYAKI”) で前処理付き GMRES 法の評価を行った。

連立一次方程式の求解手法である反復解法は、適当に選んだ初期値から出発して、真の解に収束していく近似解の列を逐次作成していく手法である。反復解法として**クリロフ (Krylov) 部分空間**に基づく**一般化最小残差 (GMRES) 法**がある。GMRES 法は右辺ベクトル $\boldsymbol{b}$ から出発してクリロフ部分空間を拡大しながら正規直交基底 $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{j+1}$ を生成し、各ステップにおいて残差ノルム $\|\boldsymbol{r}_j\|_2 = \|\boldsymbol{b} - A\boldsymbol{x}_j\|_2$ が最小になるように近似解 $\boldsymbol{x}_j$ を更新していく手法である。

クリロフ部分空間法の収束性は一般に係数行列 A の固有値分布に依存し、固有値分布が少なくかつ 1 (単位行列) に近いほど収束が早い。収束性や安定性の改善を目的として反復に入る前にあらかじめ係数行列 A に対し**オーダリング**、**前処理**を施す。

GPU 上で GMRES 法での反復中に使われる直交化について**古典グラムシュミット 2 (CGS2) 法**と**修正グラムシュミット (MGS) 法**の比較を行った。CUDA による評価の結果、GPU 上での CGS2 法は MGS 法と比べても収束性は問題なく処理時間が大幅に短いということが分かった。

不完全 LU 分解 (ILU) 前処理付き GMRES 法について ILU(0) or ILU(1) or ILU(2) の評価を行った結果、悪条件の問題の場合は手厚い前処理 (ILU(2) など) が必要になってくることがわかった。また、ILU のフィルインを許すレベルについては、非ゼロ要素数の増加量や収束性などに応じて処理時間が最短となる適切なレベルが異なることがわかった。

オーダリングについては**逆 Cuthill-McKee (RCM)** と **Nested Dissection (ND)** について GMRES 法で評価を行った。求解対象の行列によるが概ね、CPU の場合は RCM を適用することで収束性が向上し処理時間が短縮でき、GPU の場合は RCM を適用することで収束性が向上したが処理時間は伸びた。また、GPU の場合はオーダリングに ND を用いると求解対象の行列によるが概ね、GMRES の処理時間を短縮できることが分かった。