Performance Evaluation of the Mixed Precision GMRES(m) Method using FP64 and FP32

Yingqi Zhao
Hokkaido University
Japan
yqzhao2016@hotmail.com

Takeshi Fukaya
Hokkaido University, JST PRESTO
Japan
fukaya@iic.hokudai.ac.jp

Takeshi Iwashita
Hokkaido University
Japan
iwashita@iic.hokudai.ac.jp

1 INTRODUCTION

Traditionally, FLOPS (Floating-point Operations Per Second) for double-precision floating-point number (FP64) operation has been used as a metric for the performance of a computer system. However, it is getting difficult to improve FP64 FLOPS of a single processor due to the limitation of power budget. Besides, some new applications gradually accept low precision computing.

Under this circumstances, it is necessary to develop a new method that can exploit low precision computing and provide computed results with the same accuracy as by traditional methods using only FP64. The key idea here is mixed precision computing, in which both FP64 and low precision computing (e.g., FP32) are combined.

In this paper, the target problem is a system of linear equations $Ax = b$, where $A \in \mathbb{R}^n$ is large and sparse. For this problem, a mixed precision variant of the GMRES(m) method using FP64 and FP32 (MP-GMRES(m)) is discussed, and its numerical behaviour is investigated in detail through numerical experiments.

2 MIXED PRECISION GMRES(m) METHODS

The mixed precision variant of the GMRES(m) method (Algorithm 1) has the mathematical background related to the iterative refinement scheme for the solution of linear system ; this relation is clearly explained in the study by Imakura et al. [3] The effectiveness of MP-GMRES(m) has been confirmed in some cases [1, 4], however, its numerical behaviour is sill not clear.

Algorithm 1 Mixed Precision GMRES(m)

Require: x_0: initial guess, ϵ: convergence criterion, A; coefficient matrix, b: right-hand side vector, maximum number of iterations

1: $A^{(\mathrm{L})} = \mathrm{Low}(A)$ ▷ precision converts from *standard* to *low*
2: **repeat**
3: $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$
4: **if** $\beta / \|b\|_2 \leq \epsilon$ **then return** x_0
5: $v_0 = r_0 / \beta$
6: $\beta^{(\mathrm{L})} = \mathrm{Low}(\beta)$, $v_0^{(\mathrm{L})} = \mathrm{Low}(v_0)$
7: Compute $V_m^{(\mathrm{L})}$ and $\tilde{H}_m^{(\mathrm{L})}$ by the m-step Arnoldi process with $A^{(\mathrm{L})}$ and $v_0^{(\mathrm{L})}$. ▷ using *low precision* arithmetic
8: Compute $y_m^{(\mathrm{L})}$ from $\beta^{(\mathrm{L})}$ and $\tilde{H}_m^{(\mathrm{L})}$. ▷ using *low precision* arithmetic
9: $z_m^{(\mathrm{L})} = V_m^{(\mathrm{L})} y_m^{(\mathrm{L})}$ ▷ using *low precision* arithmetic
10: $z_m = \mathrm{Std}(z_m^{(\mathrm{L})})$ ▷ precision converts from *low* to *standard*
11: $x_0 = x_0 + z_m$
12: **until** attain the maximum number of iterations

Ensure: x_0

3 NUMERICAL EXPERIMENTS

In order to clarify the numerical behaviour of MP-GMRES(m) , numerical experiments with the matrices obtained from the SuiteSparse Matrix Collection [2] are conducted, in which MP-GMRES(m) is compared with conventional GMRES(m) using only FP64 (FP64-GMRES(m)). The number of iterations, the achievable accuracy and the execution time are investigated. The detail of the numerical results will be presented in the poster.

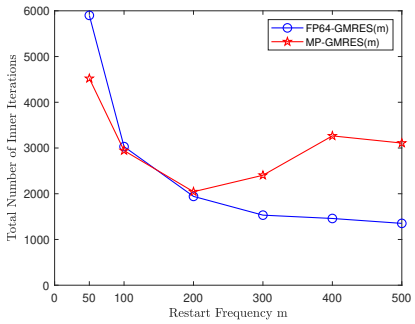

Figure 1: The change of the number of the total inner iterations in FP64-GMRES(m) and MP-GMRES(m) for `memplus`.

In terms of the number of iterations, when m is small, the number of iterations of the two methods are almost equal. However, when m becomes larger, the number of iterations of FP64-GMRES(m) basically decreases, but that of MP-GMRES(m) tends to increase. Take matrix named `memplus` for example, Figure 1 illustrates the change of the number of the total inner iterations with m.

For the achievable accuracy, both FP64-GMRES(m) and MP-GMRES(m) attained the convergence criterion ($\|b - Ax\|_2 / \|b\|_2 \leq 10^{-10}$) in most cases. If both two methods did not converge, there is almost no difference in the final accuracy.

When m is small, the execution time of both methods tends to be short. For many test matrices, when the number of iterations is almost equal, MP-GMRES(m) outperforms FP64-GMRES(m) in terms of the execution time. Even in the case that larger m reduces the number of iterations in FP64-GMRES(m), e.g., matrix named `wang4`, MP-GMRES(m) is still faster, in which about three times the iterations are required.

REFERENCES

[1] Hartwig Anzt, Vincent Heuveline, and Björn Rocker. 2011. An Error Correction Solver for Linear Systems: Evaluation of Mixed Precision Implementations. In *High Performance Computing for Computational Science – VECPAR 2010*. 58–70.
[2] Timothy A. Davis and Yifan Hu. 2011. The University of Florida Sparse Matrix Collection. *ACM Trans. Math. Software* 38, 1 (Dec. 2011), 1:1–1:25.
[3] Akira Imakura, Tomohiro Sogabe, and Shao-Liang Zhang. 2012. An Efficient Variant of the GMRES(m) Method Based on the Error Equations. *East Asian Journal on Applied Mathematics* 2, 1 (2012), 19–32.
[4] Neil Lindquist, Piotr Luszczek, and Jack Dongarra. 2022. Accelerating Restarted GMRES with Mixed Precision Arithmetic. *IEEE Transactions on Parallel and Distributed Systems* 33 (2022), 1027–1037. Issue 4.