An attempt of exploiting low precision computing in the GMRES(m) method

March 2, 2023
2023 SIAM Conference on Computational Science and Engineering (CSE23)
@RAI Congress Centre, Amsterdam, The Netherlands

Takeshi Fukaya (Hokkaido University)

Collaborator:
Yingqi Zhao and Takeshi Iwashita (Hokkaido University)

Introduction

◆Problem setting

Solving a linear system with a sparse coefficient matrix:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

A: n-dimensional sparse matrix
(regular, real and not symmetric)

◆Target algorithm

GMRES(m) method: restarted GMRES method

◆Goal

Through numerical experiments, to investigate possibilities of exploiting low precision computing in the GMRES(m) method without the loss of accuracy of obtained solutions (baseline: GMRES(m) with FP64 only).

Key idea

◆Iterative refinement (IR) & mixed precision

Step 1. computing residual: $\boldsymbol{r}_k := \boldsymbol{b} - A\boldsymbol{x}_k$

Step 2. solving error equation: $\boldsymbol{e}_k = A^{-1}\boldsymbol{r}_k$ (solving a linear system)

Step 3. updating the solution: $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \boldsymbol{e}_k$

In Step 2, low precision computing can be acceptable.

◆Relation between IR and GMRES(m)

1: **repeat**

2: Solve $A\boldsymbol{x} = \boldsymbol{b}$ by m-iteration GMRES with the initial guess $\boldsymbol{x}_0$,
and find the solution $\boldsymbol{x}_m$.

3: $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}_m$ (update the initial guess)

4: **until** satisfy required accuracy condition or attain maximum iteration number

GMRES(m) has the structure of IR (IR using m-step GMRES for Step 2).

GMRES(m) using low precision computing

Input: An initial guess $\boldsymbol{x}_0$

1: **repeat**

2: $\boldsymbol{r}_0 \leftarrow \boldsymbol{b} - A\boldsymbol{x}_0, \quad \beta \leftarrow \|\boldsymbol{r}_0\|_2$

3: $\boldsymbol{v}_0 \leftarrow \boldsymbol{r}_0/\beta$

4: Compute m-step Arnoldi process with A and $\boldsymbol{v}_0$, and get V_m and $\overline{H}_m$.

5: Compute $\boldsymbol{y}_m$ from β and $\overline{H}_m$.

6: $\boldsymbol{e}_m \leftarrow V_m\boldsymbol{y}_m$

7: $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}_0 + \boldsymbol{e}_m$

8: **until** satisfy required accuracy condition or attain maximum iteration number

Check convergence here (using FP64)

corresponds to Step 2 in IR (solving error equation)

Low precision computing can be acceptable.

◆ What we present in this talk

Numerical results of two attempts of introducing low precision computing:

- GMRES(m) using FP32 and FP64
- GMRES(m) using low precision data including those lower than FP32.

GMRES(m) using FP32 & FP64

Outline of the algorithm

Input: An initial guess $\boldsymbol{x}_0$

1: $A^{(\mathrm{FP32})} \leftarrow \mathrm{ToFP32}(A)$ — Prepare matrix data in FP32

2: **repeat**

3: $\boldsymbol{r}_0 \leftarrow \boldsymbol{b} - A\boldsymbol{x}_0, \quad \beta \leftarrow \|\boldsymbol{r}_0\|_2$

4: $\boldsymbol{v}_0^{(\mathrm{FP32})} \leftarrow \mathrm{ToFP32}(\boldsymbol{r}_0/\beta), \quad \beta^{(\mathrm{FP32})} \leftarrow \mathrm{ToFP32}(\beta)$ — convert to FP32 data

5: Compute m-step Arnoldi process in low precision with $A^{(\mathrm{FP32})}$ and $\boldsymbol{v}_0^{(\mathrm{FP32})}$, and get $V_m^{(\mathrm{FP32})}$ and $\overline{H}^{(\mathrm{FP32})}{}_m$.

6: Compute in low precision $\boldsymbol{y}_m^{(\mathrm{FP32})}$ from $\beta^{(\mathrm{FP32})}$ and $\overline{H}^{(\mathrm{FP32})}{}_m$.

7: $\boldsymbol{e}_m^{(\mathrm{FP32})} \leftarrow V_m^{(\mathrm{FP32})} \boldsymbol{y}_m^{(\mathrm{FP32})}$

8: $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}_0 + \mathrm{ToFP64}(\boldsymbol{e}_m^{(\mathrm{FP32})})$ — convert to FP64 data

9: **until** satisfy required accuracy condition or attain maximum iteration number

We focus on the numerical behavior (convergence property) of the mixed-precision GMRES(m) method using FP32 and FP64 compared with that of GMRES(m) using only FP64. (For some problems, its effectiveness in execution time has been already reported.)

Evaluation on attainable accuracy

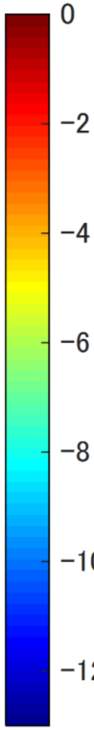
$\log_{10} \frac{\|\boldsymbol{b} - A\boldsymbol{x}\|_2}{\|\boldsymbol{b}\|_2}$ at the maximum iterations (or convergence condition)

GMRES(m) using only FP64

matrix \ m	50	100	200	300	400	500
FEM_3D_thermal1	−13	−13	−13	−13	−13	−13
Zhao1	−12	−13	−13	−13	−13	−13
ns3Da	−13	−13	−13	−13	−13	−13
poisson3Da	−13	−13	−13	−13	−13	−13
epb1	−13	−13	−13	−13	−13	−13
light_in_tissue	−13	−13	−13	−13	−13	−13
coupled	−5	−13	−13	−13	−13	−13
Zhao2	−1	−13	−13	−13	−13	−13
waveguide3D	−4	−5	−7	−11	−12	−13
memplus	−13	−13	−13	−13	−13	−13
wang4	−1	−1	−13	−13	−13	−13
viscoplastic2	−1	−13	−13	−13	−13	−13
inlet	−1	−1	−1	−1	−1	−1
airfoil_2d	−8	−9	−10	−9	−10	−11
chipcool1	−1	−3	−13	−13	−13	−13
garon2	−1	−2	−3	−3	−3	−3
sme3Da	−1	−1	−2	−2	−2	−4
TSOPF_RS_b39_c7	−1	−1	−13	−13	−13	−13

MP-GMRES(m) using FP32 & FP64

matrix \ m	50	100	200	300	400	500
FEM_3D_thermal1	−13	−13	−13	−13	−13	−13
Zhao1	−13	−13	−13	−13	−13	−12
ns3Da	−13	−13	−13	−13	−13	−13
poisson3Da	−13	−13	−13	−13	−13	−13
epb1	−13	−13	−13	−13	−13	−13
light_in_tissue	−13	−13	−13	−13	−13	−13
coupled	−8	−13	−13	−13	−13	−13
Zhao2	−1	−13	−13	−13	−13	−13
waveguide3D	−3	−5	−6	−12	−12	−13
memplus	−13	−13	−13	−13	−13	−13
wang4	−1	−1	−13	−13	−13	−13
viscoplastic2	−1	−13	−13	−13	−13	−13
inlet	−1	−1	−1	−1	−1	−1
airfoil_2d	−3	−3	−3	−3	−11	−7
chipcool1	−1	−6	−13	−12	−12	−12
garon2	−1	−2	−3	−3	−3	−3
sme3Da	−1	−1	−2	−2	−3	−7
TSOPF_RS_b39_c7	−1	−1	−1	−1	−1	−1



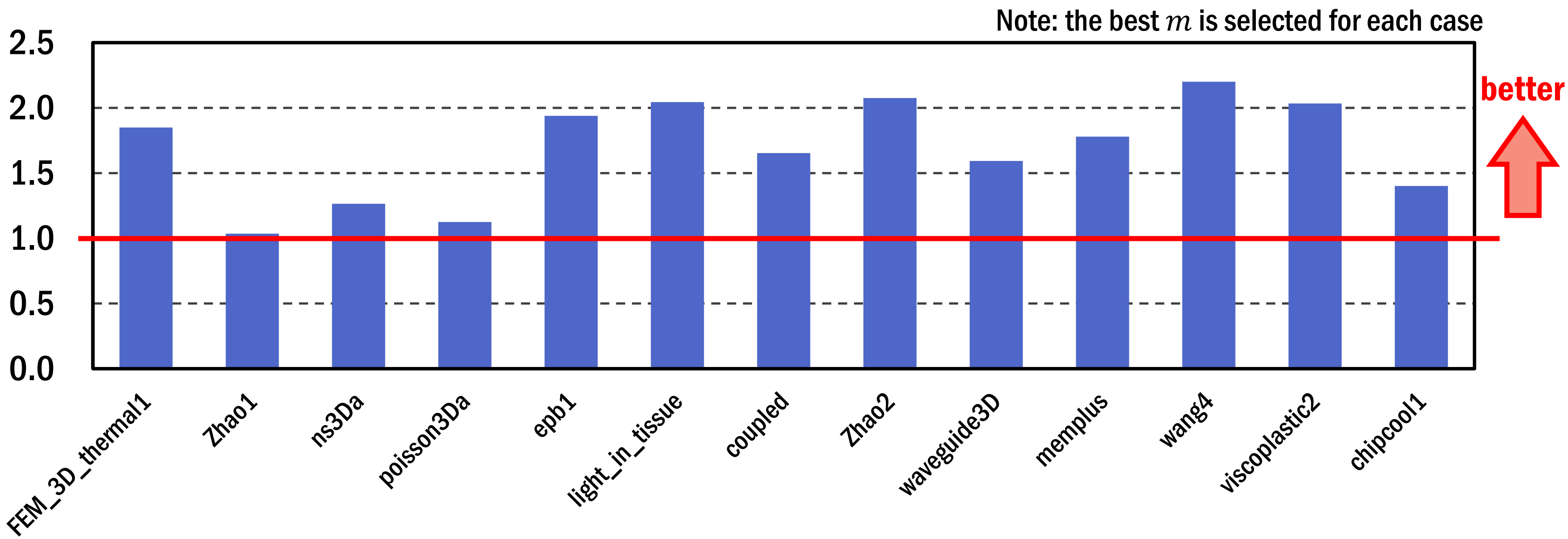
If a problem can be solved by GMRES(m) using only FP64, the problem is expected to be solved also by MP-GMRES(m) using FP32 & FP64 (with the almost same # of iterations).

Evaluation on execution time

	Both converged	Only FP64 converged	Both not converged
# of matrices	13	1	4

Speedup of MP-GMRES(m) using PF32 & FP64 over FP64 GMRES(m)

(by a standard thread parallel implementation, on a system with 2x 20-core Skylake Xeon)



For details, please see our paper: Y. Zhao et al., Numerical Investigation into the Mixed Precision GMRES(m) Method Using FP64 and FP32, JIP, 30 (2022), 525-537 (Open access).

GMRES(m) using low precision data including those lower than FP32

Apology

Unpublished results

Conclusion

Conclusion

◆ Summary

Through numerical experiments, we investigated possibilities of introducing low precision computing into the GMRES(m) method.

- The MP-GMRES(m) using FP32 and FP64 shows the similar convergence property as that of GMRES(m) using only FP64.
- There is a considerable possibility of introducing lower precision data than FP32 into the GMRES(m) method if a problem is not difficult.
- The impact of reducing the precision of A and V is different; more aggressive reduction for A will be acceptable than for V.

◆ Future work

- Further numerical experiments
- Theoretical analysis
- Discussion on expected speedup (e.g., performance modeling)
- Study on the case of using preconditioners