A challenge of exploiting low precision computing in iterative linear solvers

March 6, 2023
HPC challenges for new extreme scale applications
@Hôtel Pullman Paris Montparnasse, Paris, France

Takeshi Fukaya (Hokkaido University)

Collaborator:
Yingqi Zhao and Takeshi Iwashita (Hokkaido University)

Introduction

Background & Goal

◆Background

Mixed precision algorithms using low precision computing is a research topic actively studied in the fields of HPC and Applied Mathematics.

- Difficulty in the improvement of FLOPS of double precision (FP64)
- Appearance of hardware specialized in low precision computing (FP32/FP16) under the demand in AI applications

◆Goal

Developing efficient sparse linear solvers

- that can exploit low precision computing (mixed precision computing)
- that can provide solutions as accurate as by the conventional solvers using FP64
  (for the easy use in applications without additional validation)

Overview of this talk

◆Problem setting

Solving a linear system with a sparse matrix:

$$Ax = b$$

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

◆Target algorithm

GMRES(m) method: restarted GMRES method

◆Objective

Through numerical experiments, to investigate possibilities of introducing low precision computing in the GMRES(m) method:

- using FP32 and FP64
- using low precision data including those lower than FP32

Mixed precision GMRES(m) using low precision computing

GMRES and GMRES(m) [Saad & Schultz, 1986]

◆GMRES: Generalized Minimal RESidual method

$\boldsymbol{x}_0$: an any initial guess, and find

Krylov subspace

$$\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k(A, \boldsymbol{r}_0), \quad \mathcal{K}_k(A, \boldsymbol{r}_0) \coloneqq \mathrm{span}\{\boldsymbol{r}_0, A\boldsymbol{r}_0, \dots, A^{k-1}\boldsymbol{r}_0\}$$

so as to minimize $\|\boldsymbol{r}_k\|_2$, where $\boldsymbol{r}_i = \boldsymbol{b} - A\boldsymbol{x}_i \ (i = 0, 1, \dots, k)$.

Need of avoiding larger k because required memory and #flops/iteration $\propto k$.

◆GMRES(m): restarted GMRES

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

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

Iterative refinement & mixed precision

◆Iterative refinement (IR) for solving linear system

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: $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \boldsymbol{e}_k$

◆IR-based mixed precision computing

- Solution ($\boldsymbol{x}_k$) is stored in standard precision.
- Computing residual (Step 1) by using standard precision.
- Solving error equation (Step 2) by using low precision.

◆Related studies based on LU factorization

- Exploiting half precision (fp16) in GPU: A. Haidar et al., SC18.
- Theoretical analysis of IR using three precisions: E. Carson et al., SISC, 2018.

Relation between GMRES(m) & IR

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

3: $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}_m$

Mathematically equivalent (e.g., A. Imakura et al., 2012)

2: Solve $A\boldsymbol{e} = \boldsymbol{r}_0$ by m-iteration GMRES with the initial guess $\boldsymbol{e}_0 := \boldsymbol{0}$, and find the solution $\boldsymbol{e}_m$, where $\boldsymbol{r}_0 := \boldsymbol{b} - A\boldsymbol{x}_0$.

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

GMRES(m) has the structure of iterative refinement.

(Inner computation of GMRES(m) can accept lower precision computing.)

GMRES(m) using low precision computing

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

Check convergence here (using FP64)

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

corresponds to Step 2 in IR (solving error equation)

Low precision computing can be acceptable.

◆ What we investigate

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.)

Settings in numerical experiments

◆Environment: Grand Chariot @ Hokkaido Univ.

- CPU: Intel Xeon Gold 6148 (Skylake, 20-core, 2.4GHz) x 2
- Program: C language + OpenMP, using CRS format for sparse matrix
- Compiler: icc ver. 18.0.3 with "–O3 –qopenmp –ipo –xCORE-AVX512"
- # of threads: 40 (affinity: compact)

◆Experimental settings

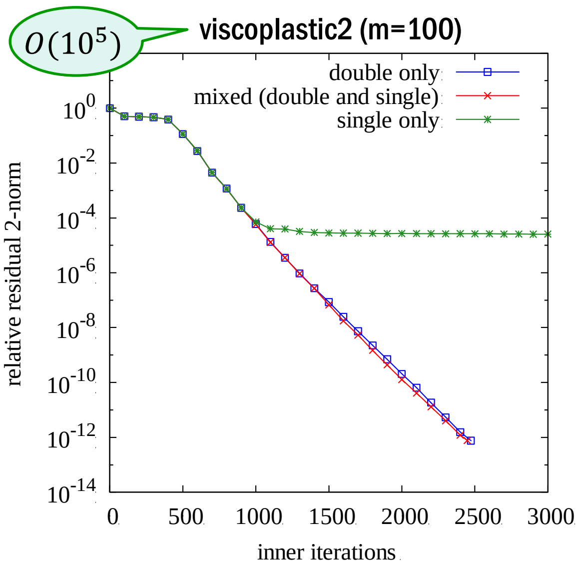
- Test matrix: selected from SuiteSparse Matrix Collection (Florida Univ.)
- $\boldsymbol{b} = (1,1,\ldots,1)^{\top}$ and $\boldsymbol{x}_0 = \boldsymbol{0}$ at beginning.
- Convergence criterion: $\|\boldsymbol{b} - A\boldsymbol{x}\|_2 / \|\boldsymbol{b}\|_2 \leq 10^{-12}$
- Maximum iteration number: total # of inner iterations attains 50,000.
- m: 50, 100, 200, 300, 400, 500

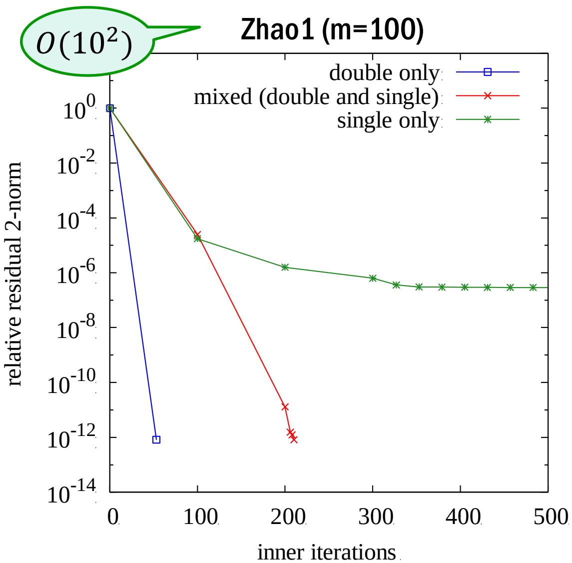
List of matrices used in experiments

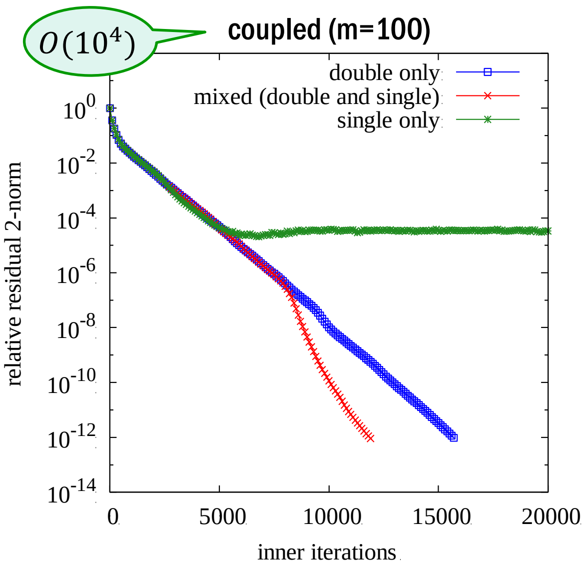
Cond. num.	Matrix name	size (n)	nnz	application
$O(10^2)$	FEM_3D_thermal1	17,880	430,740	Thermal Problem
	Zhao1	33,861	166,453	Electromagnetics Problem
	ns3Da	20,414	1,679,599	Computational Fluid Dynamics Problem
$O(10^3)$	poisson3Da	13,514	352,762	Computational Fluid Dynamics Problem
	epb1	14,734	95,053	Thermal Problem
	light_in_tissue	29,282	406,084	Electromagnetics Problem
$O(10^4)$	coupled	11,341	97,193	Circuit Simulation Problem
	Zhao2	33,861	166,453	Electromagnetics Problem
	waveguide3D	21,036	303,468	Electromagnetics Problem
$O(10^5)$	memplus	17,758	99,147	Circuit Simulation Problem
	wang4	26,068	177,196	Semiconductor Device Problem
	viscoplastic2	32,769	381,326	Materials Problem
$O(10^6)$	inlet	11,730	328,323	Model Reduction Problem
	airfoil_2d	14,214	259,688	Computational Fluid Dynamics Problem
	chipcool1	20,082	281,150	Model Reduction Problem
$O(10^7)$	garon2	13,535	373,235	Computational Fluid Dynamics Problem
	sme3Da	12,504	874,887	Structural Problem
	TSOPF_RS_b39_c7	14,098	252,446	Power Network Problem

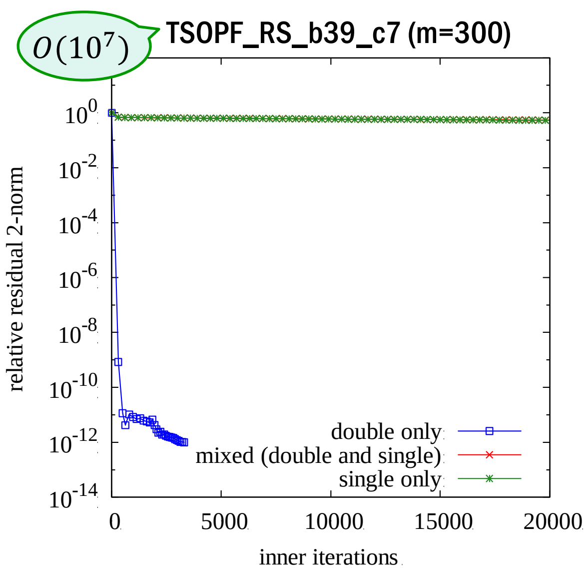
Examples of convergence history

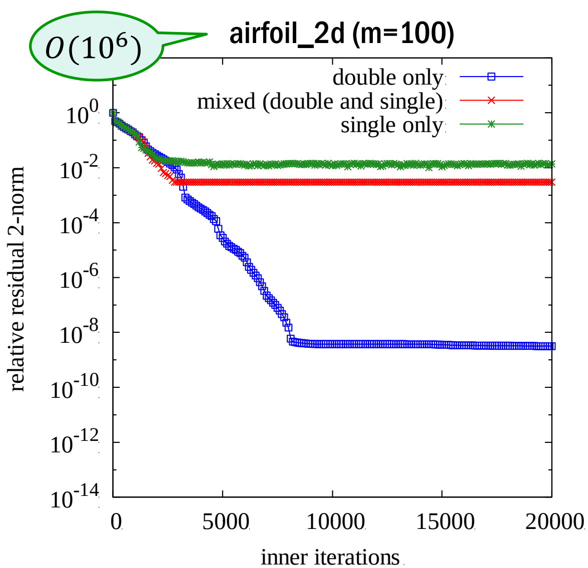
Note: at each restart in each algorithm, computing the residual norm in double precision

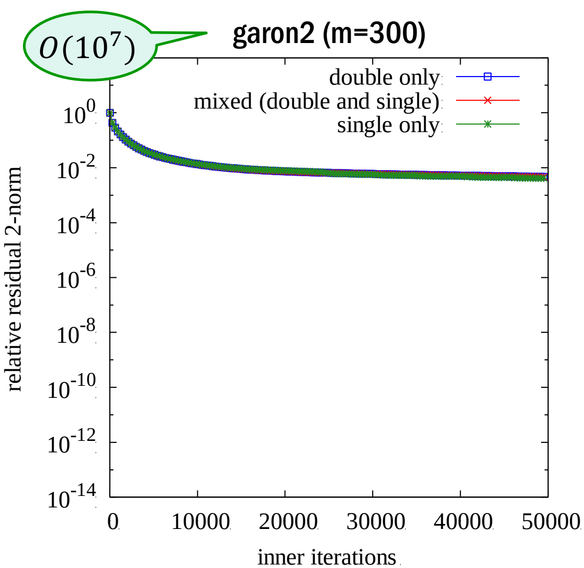












Evaluation on attainable accuracy

$\log_{10} \frac{\|\boldsymbol{b} - A\boldsymbol{x}\|_2}{\|\boldsymbol{b}\|_2}$ **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

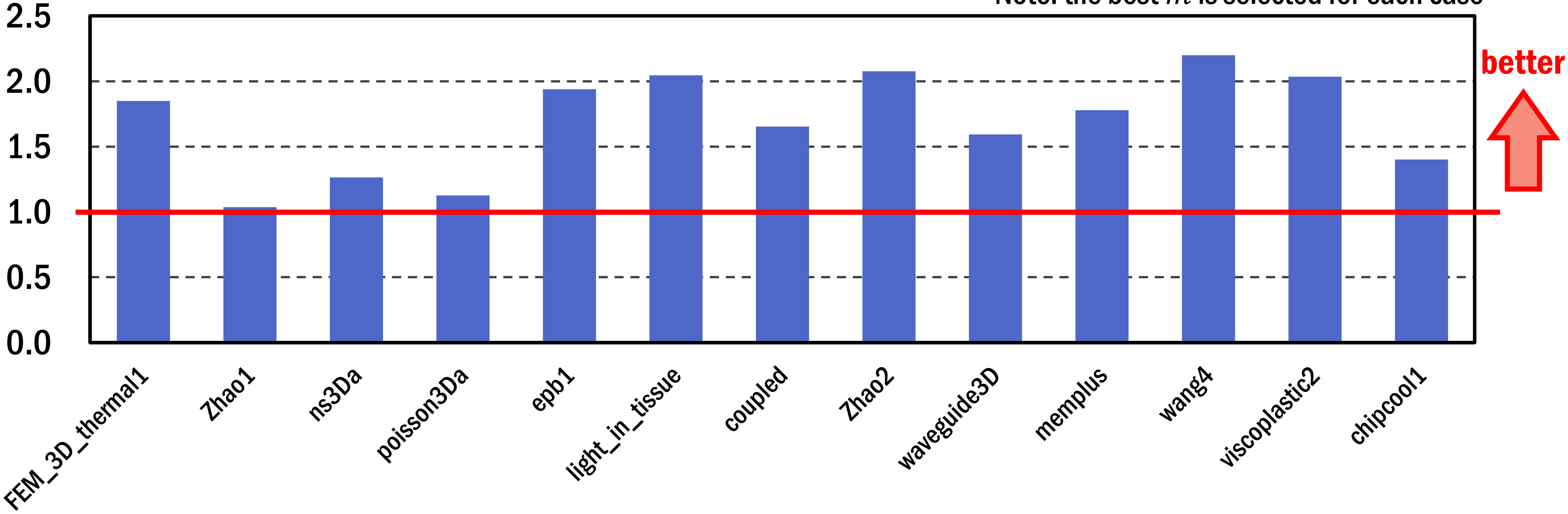
If a problem can be solved by GMRES(m) using only FP64, the problem is expected to be solved also by MP-GMRES(m) using FP32 & FP64 (excepting only a few cases).

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)

Note: the best m is selected for each case



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