Effectiveness of Partial Use of Quadruple Precision Arithmetic with QuPAT to Iterative Methods

Tsubasa Saito
(Tokyo University of Science, Japan)
Emiko Ishiwata
(Tokyo University of Science, Japan)
Hidehiko Hasegawa
(University of Tsukuba, Japan)

ICIAM 2011 July 19 2011

Background & Motivation

- ✓ Iterative methods may not converge because of round-off errors.

- ✓ We apply quadruple precision arithmetic **with QuPAT** to iterative methods.
- ✓ To improve convergence with relatively small cost, we use quadruple precision arithmetic **partially**.

Background & Motivation

- ✓ In floating point arithmetic, we cannot avoid :
  - ✓ Round-off errors
  - ✓ Cancellation
  - ✓ Information loss
- ✓ It is difficult to implement multiple precision arithmetic without any special hardware.

A **convenient** quadruple precision arithmetic environment **QuPAT**[1] on **Scilab** has developed.

[1] T. Saito, E. Ishiwata and H. Hasegawa, Development of quadruple precision arithmetic toolbox qupat on scilab, ICCSA2010, Proceedings Part II, LNCS 6017, pp. 60-70, Springer (2010).

Agenda

- Background & Motivation
- DD and QD arithmetic
- Features of QuPAT
  (Quadruple Precision Arithmetic Toolbox)
- Numerical experiment
  (Partial use of DD arithmetic)
- Conclusion

Multiple precision arithmetic

DD and QD arithmetic (Hida, Li and Bailey [2])

- ✓Using **some double precision numbers**
  - ✓DD : quasi quadruple precision
  - ✓QD : quasi octuple precision
- ✓**Combination** of double precision arithmetic operations

  ⟶ **only** needs **double** precision arithmetic environment

QuPAT uses these arithmetic on Scilab

[2] Y. Hida, X. S. Li and D. H. Bailey, Quad-double arithmetic: algorithms, implementation, and application. Technical Report LBNL-46996, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 (2000).

Number representation

DD number a is represented by 2 double precision numbers as follow:

$$a = a_0 + a_1, \quad \begin{cases} a_0 : \text{higher part of } a \\ a_1 : \text{lower part of } a \end{cases}$$

where a_0 and a_1 satisfy $|a_1| \leq \frac{1}{2}\mathrm{ulp}(a_0)$.

*ulp (units in the last place)

	a_0			a_1		
DD	s	$e_1 \dots e_{11}$	$m_1 \dots m_{52}$	s	$e_1 \dots e_{11}$	$m_1 \dots m_{52}$
IEEE quad.	s	$e_1 \dots e_{15}$	$m_1 \dots m_{112}$			

QD number is represented in the same way as follow:

$$a = a_0 + a_1 + a_2 + a_3$$

Number of operations

		+, −	∗	/	total
DD	addition, subtraction	11	0	0	11
	multiplication	15	9	0	24
	division	17	8	2	27
QD	addition, subtraction	84	0	0	84
	multiplication	163	46	0	209
	division	713	88	5	806

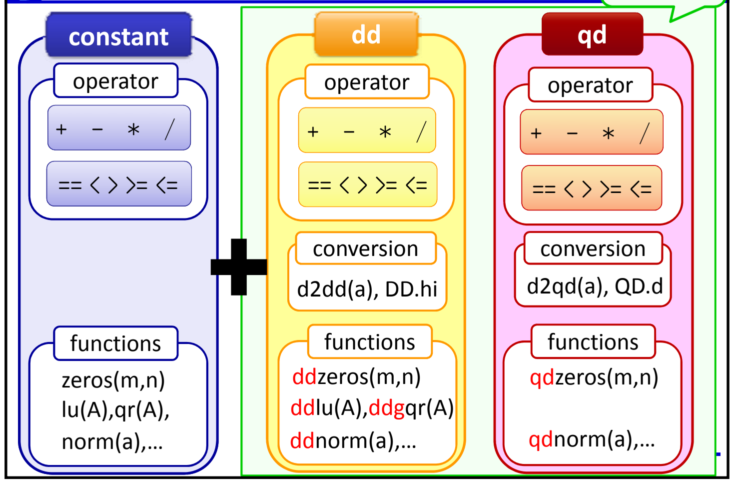
QuPAT(Quadruple Precision Arithmetic Toolbox)

Convenient multiple precision arithmetic toolbox on Scilab

- ✓The same operator (+, −, ∗, /) can be used for double, DD, and QD arithmetic.

  ⇨ We can write a code simply and easily.
- ✓Double, DD, and QD arithmetic can be used at the same time, and also mixed precision arithmetic is available.
- ✓It is independent of any hardware and operating systems.

Relationship between Scilab & QuPAT



Agenda

- Background & Motivation
- DD and QD arithmetic
- Features of QuPAT
  (Quadruple Precision Arithmetic Toolbox)
- **Numerical experiment**
  (Partial use of DD arithmetic)
- Conclusion

9

Application of QuPAT

GCR (Generalized Conjugate Residual) method

One of the Krylov subspace method for solving nonsymmetric linear systems $\boldsymbol{Ax} = \boldsymbol{b}$.

Properties

n : dimension of A

- In theory, the residual norm converges at most n iterations.
- Using floating point arithmetic, the residual norm often stagnates by round-off errors.

10

Computational condition

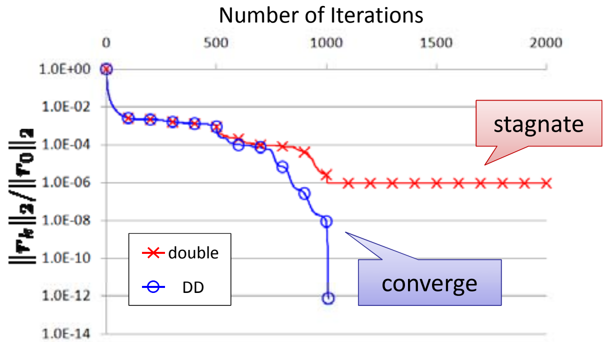
Matrix : olm2000
Dimension : 2000
Condition number : 5.94×10^6
(from The University of Florida Sparse Matrix Collection)

Scilab version 5.1.1 on Windows XP

Maximum iteration : 2000
Initial vector : $\boldsymbol{x_0} = (0, 0, \cdots, 0)^T$
Solution vector : $\boldsymbol{x^*} = (1, 1, \cdots, 1)^T$
Stopping criterion : $\|\boldsymbol{r_k}\|_2 \leq 10^{-12} \|\boldsymbol{r_0}\|_2$

11

Computational results of double and DD



	Number of Iter.	$\|b-Ax\|_2/\|r_0\|_2$	$\|x-x^*\|_\infty/\|x^*\|_\infty$
double	×	9.11E-07	8.67E-03
DD	1009	7.10E-13	3.56E-09

12

Reviewing iterative process of GCR

Let x_0 be an initial guess.
set $r_0 = b - Ax_0,\ p_0 = r_0,\ q_0 = Ap_0,\ k = 0$
while $\|r_k\|_2 < \varepsilon\|r_0\|_2$ and $k < n$ do
$\alpha_k = (r_k, q_k)/(q_k, q_k)$
$x_{k+1} = x_k + \alpha_k p_k$
$r_{k+1} = r_k - \alpha_k q_k$
for $i = 0, \cdots, k$ do
$\beta_{k,i} = -(Ar_{k+1}, q_i)/(q_i, q_i)$
$p_{k+1} = r_{k+1} + \sum_{i=0}^{k} \beta_{k,i} p_i$
$q_{k+1} = Ar_{k+1} + \sum_{i=0}^{k} \beta_{k,i} q_i$
$k = k + 1$

iteration

Updating approximate solution and residual (Strategy 1)

Generating a basis vector (Strategy 2)

DD arithmetic is partially used for α and β [3].

[3] T. Saito, E. Ishiwata and H. Hasegawa, Analysis of the GCR method with mixed precision arithmetic using QuPAT, Journal of Computational Science, in press

13

Strategy 1 (using DD arithmetic for α)

Let x_0 be an initial guess.
set $r_0 = b - Ax_0,\ p_0 = r_0,\ q_0 = Ap_0,\ k = 0$
while $\|r_k\|_2 < \varepsilon\|r_0\|_2$ and $k < n$ do
$\alpha_k = (r_k, q_k)/(q_k, q_k)$
$x_{k+1} = x_k + \alpha_k p_k$
$r_{k+1} = r_k - \alpha_k q_k$
for $i = 0, \cdots, k$ do
$\beta_{k,i} = -(Ar_{k+1}, q_i)/(q_i, q_i)$
$p_{k+1} = r_{k+1} + \sum_{i=0}^{k} \beta_{k,i} p_i$
$q_{k+1} = Ar_{k+1} + \sum_{i=0}^{k} \beta_{k,i} q_i$
$k = k + 1$

- α_k (inner product, division)
  x_{k+1}, r_{k+1} (product, sum)
  …DD arithmetic
- α_k, r_{k+1}
  …DD numbers

14

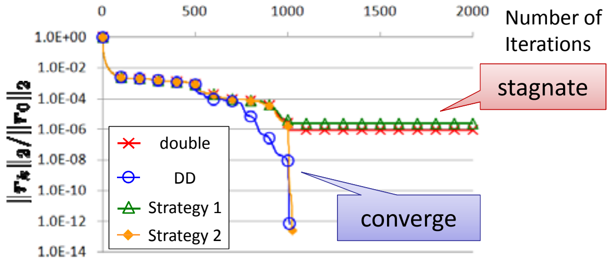
Strategy 2 (using DD arithmetic for β)

Let x_0 be an initial guess.
set $r_0 = b - Ax_0,\ p_0 = r_0,\ q_0 = Ap_0,\ k = 0$
while $\|r_k\|_2 < \varepsilon\|r_0\|_2$ and $k < n$ do
$\alpha_k = (r_k, q_k)/(q_k, q_k)$
$x_{k+1} = x_k + \alpha_k p_k$
$r_{k+1} = r_k - \alpha_k q_k$
for $i = 0, \cdots, k$ do
$\beta_{k,i} = -(Ar_{k+1}, q_i)/(q_i, q_i)$
$p_{k+1} = r_{k+1} + \sum_{i=0}^{k} \beta_{k,i} p_i$
$q_{k+1} = Ar_{k+1} + \sum_{i=0}^{k} \beta_{k,i} q_i$
$k = k + 1$

- $\beta_{k,i}$ (inner product, division)
  p_{k+1}, q_{k+1} (product, sum)
  …DD arithmetic
- $\beta_{k,i}, q_{k+1}$
  …DD numbers

15

Computational results of 2 strategies



	Number of Iter.	$\|b - Ax\|_2/\|r_0\|_2$	$\|x - x^*\|_\infty/\|x^*\|_\infty$
double	×	9.11E-07	8.67E-03
DD	1009	7.10E-13	3.56E-09
Strategy 1	×	2.38E-06	2.28E-02
Strategy 2	1025	2.87E-11	5.92E-09

Strategy 2 shows almost the same convergence as DD !

Relation between α_k and convergence

Let x_0 be an initial guess.

set $r_0 = b - Ax_0,\ p_0 = r_0,\ q_0 = Ap_0,\ k = 0$

while $\|r_k\|_2 < \varepsilon\|r_0\|_2$ and $k < n$ do

$\alpha_k = (r_k, q_k) / (q_k, q_k)$ ← $\alpha_k \approx 0$ ⇒ $r_{k+1} \approx r_k$

$x_{k+1} = x_k + \alpha_k p_k$

$r_{k+1} = r_k - \alpha_k q_k$

for $i = 0, \cdots, k$ do

$\beta_{k,i} = -(Ar_{k+1}, q_i) / (q_i, q_i)$

$p_{k+1} = r_{k+1} + \sum_{i=0}^{k} \beta_{k,i} p_i$

$q_{k+1} = Ar_{k+1} + \sum_{i=0}^{k} \beta_{k,i} q_i$

$k = k + 1$

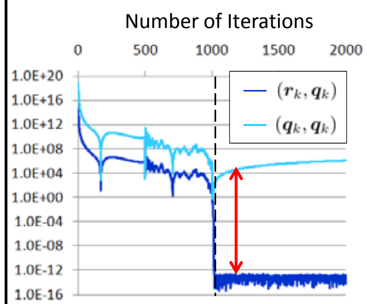
iteration

α_k is reflected by the accuracy of q_k

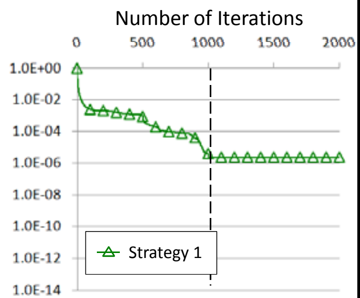
17

The reason of stagnation is α_k

$|\alpha_k|$ in Strategy 1



Convergence history



$\alpha_k \approx 0$ → Stagnate

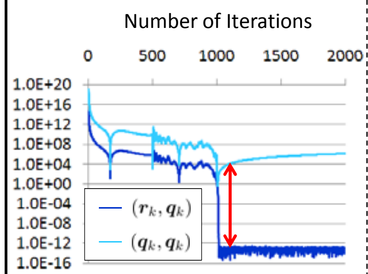
$$\alpha_k = \frac{(r_k, q_k)}{(q_k, q_k)}$$

18

Compare two strategies

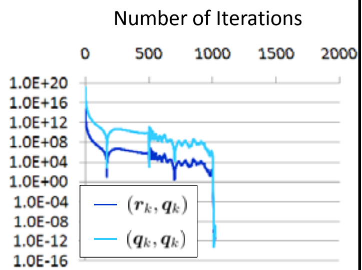
$$\alpha_k = \frac{(r_k, q_k)}{(q_k, q_k)}$$

Strategy 1



$\alpha_k \approx 0$ → Stagnate

Strategy 2



Maintain convergence

19

Conclusion

- Partial use of DD arithmetic to GCR is effective!
  - Using DD for only $\beta_{k,i}$ and q_k achieves **almost the same convergence** as full DD
- QuPAT enable us to write multiple precision code **simply** on Scilab!
  - Can be used **the same operators** $(+, -, *, /)$
  - **Independent** of any hardware and operating systems

QuPAT is available at
http://www.mi.kagu.tus.ac.jp/qupat.html

20