Mixed-precision Iterative Refinement

Ned Nedialkov

McMaster University

9 March 2023

Outline

Introduction

- Mixed-precision (MP) algorithms use two or more precisions from
  - **half**, **single** and **double** precisions in hardware, and
  - **quadruple** precision (usually) in software.
- Perform most of the work in lower precision.
  Do the rest in higher precision to improve accuracy.
- Exploiting lower precision(s):
  - faster floating-point computations
  - less: storage, data movement, communications, energy consumption

precision	bits	range	unit roundoff
bfloat16	16	$10^{\pm 38}$	4×10^{-3}
half	16	$10^{\pm 5}$	5×10^{-4}
single	32	$10^{\pm 38}$	6×10^{-8}
double	64	$10^{\pm 308}$	1×10^{-16}
quadruple	128	$10^{\pm 4932}$	1×10^{-34}

- Modern GPUs
  - **half**, **bfloat16** much faster than **single**
  - **single** 2x faster than **double**
    e.g. Nvidia A100 half:single 4x, single:double 2x
- Modern CPUs
  - **single** can be 2x faster than **double**
- **quadruple**
  a good software implementation $\approx$ 10x slower than **double**

Iterative refinement

Solve $Ax = b$ where A is a nonsingular $n \times n$ matrix, b is an n vector.

Iterative refinement (IR), Wilkinson 1963.

- Compute an approximate solution x_0.
- Residual is
$$r = b - Ax_0 = A \underbrace{(x - x_0)}_{d}.$$
- Solve $Ad = r$.
- Update $x_1 = x_0 + d$.
- Repeat until
  - max number of iterations is reached or
  - a stopping criterion is satisfied.

Key:

- Evaluate the residual in higher than working precision, to reduce cancellations.
- Do LU factorization (most expensive part) in lower precision.

LU-IR in 3 precisions

unit roundoff	precision
u	working
u_{fac}	for LU factorization and solving $Ax = b$, $Ad = r$
u_{res}	for evaluating residual $b - Ax$

Various combinations

$(u_{\mathsf{fac}}, u, u_{\mathsf{res}}) = (\text{half}, \text{single}, \text{double}), (\text{single}, \text{double}, \text{quad})$, etc.

Algorithm 1 (LU-IR in 3 precisions).

A and b are in working precision u
LU factorization of A u_{fac} most expensive
solve $Ay = b$ with LU u_{fac}
store y in x in u
for $i = 1$: maxiter or until converged
  compute $b - Ax$ u_{res}
  store result in r in u_{fac}
  solve $Ad = r$ using LU u_{fac}
  update $x = x + d$ u

- Fixed precision IR: all precisions are the same.
- Traditional IR: $u_{\text{fac}} = u$, e.g. u is single, u_{res} is double.
- If LU is computed by Gauss elimination with partial pivoting (GEPP), convergence if $\kappa(A)u_{\text{fac}} < 1$, $\kappa(A) = \|A\|_\infty \cdot \|A^{-1}\|_\infty$

	u_{fac}	u	u_{res}	$\max \kappa(A)$	$\|\widehat{x} - x\|/\|x\|$
Fixed	D	D	D	10^{16}	$\kappa(A) \cdot 10^{-16}$
Traditional	D	D	Q	10^{16}	10^{-16}
LU-IR	S	D	D	10^{8}	$\kappa(A) \cdot 10^{-16}$
	S	D	Q	10^{8}	10^{-16}

S single, D double, Q quad, $\widehat{x}$ computed solution.

(From T. Mary, Exploiting Mixed Precision Arithmetic in the Solution of Linear Systems)

GMRES-IR

- S. Rump (1990) noticed: if $\widehat{L}$ and $\widehat{U}$ are computed LU factors by GEPP in u_{fac} then

$$\kappa(\widehat{U}^{-1}\widehat{L}^{-1}A) \approx 1 + \kappa(A)u_{\mathsf{fac}}.$$

  even for $\kappa(A)u_{\mathsf{fac}} \gg 1$. If u_{fac} is single, $\kappa(A) \gg 10^8$.
- GMRES-IR: when solving $Ad = r$, apply GMRES to

$$\widetilde{A}d = (\widehat{U}^{-1}\widehat{L}^{-1}A)d = \widehat{U}^{-1}\widehat{L}^{-1}r.$$

- Typically $\kappa(\widetilde{A}) \ll \kappa(A)$.
- d can be accurate even for numerically singular A.
- Convergence condition improves from

  $\kappa(A)u_{\mathsf{fac}} < 1$ in LU-IR to $\kappa(\widetilde{A})u < 1$ in GMRES-IR.

	u_{fac}	u	u_{res}	$\max \kappa(A)$	$\|\widehat{x} - x\|/\|x\|$
LU-IR	S	D	Q	10^8	10^{-16}
GMRES-IR	S	D	Q	10^{16}	10^{-16}
LU-IR	H	D	Q	10^3	10^{-16}
GMRES-IR	H	D	Q	10^{11}	10^{-16}

(From T. Mary, Exploiting Mixed Precision Arithmetic in the Solution of Linear Systems)

Sparse direct solvers

A is large and sparse.

1. Symbolic factorization
   - Determine in what order to do Gauss elimination to reduce fill-in, i.e. the number of nonzeros in the LU factors.
   - Integer arithmetic, no floating-point.
2. Numerical factorization
   - Typically the most expensive part.
   - Here reduced precision can help performance.
3. Solution by substitutions
   - Typically the least expensive part.

MP in sparse solvers

From M. Zounon, N. Higham, C. Lucas, F. Tisseur. Performance impact of precision reduction in sparse linear systems solvers

- Investigated the performance of solving $Ax = b$ in single and double.
- Speedup of 2x obtained on very large test problems.
- Subnormal numbers
  - can appear in conversion from double to single and in LU
  - expensive to handle
  - they suggest flushing subnormals to zero
- Symbolic factorization
  - if in parallel, usually does not scale very well.

For a comprehensive survey see N . Higham, T. Mary. Mixed Precision Algorithms in Numerical Linear Algebra

See also N. Higham, What is Iterative Refinement?