Block low-rank compression in mixed precision for the solution of sparse linear systems
Matthieu Gerest, Patrick Amestoy, Olivier Boiteau, Alfredo Buttari, Fabienne Jézéquel, Jean-Yves L’Excellent, Théo Mary

To cite this version:
Matthieu Gerest, Patrick Amestoy, Olivier Boiteau, Alfredo Buttari, Fabienne Jézéquel, et al.. Block low-rank compression in mixed precision for the solution of sparse linear systems. Sparse Days conference 2022, Jun 2022, Saint-Girons, France. . hal-03780548

HAL Id: hal-03780548
https://hal.science/hal-03780548
Submitted on 19 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Block low-rank compression in mixed precision for the solution of sparse linear systems

Matthieu Gerest, Cifre PhD student (EDF R&D, LIP6)
Joint work with P.Amestoy, O.Boiteau, A.Buttari, F.Jézéquel, J.-Y.L’Excellent, T.Mary

**LOW-PRECISION ARITHMETICS**

<table>
<thead>
<tr>
<th>Mantissa (m)</th>
<th>Exponent</th>
<th>Range</th>
<th>( u = 2^{-m-1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>fp64 (double)</td>
<td>52 bits</td>
<td>11 bits</td>
<td>( 10^{-308} ) = ( 1 \times 10^{-16} )</td>
</tr>
<tr>
<td>fp32 (single)</td>
<td>23 bits</td>
<td>8 bits</td>
<td>( 10^{-238} ) = ( 6 \times 10^{-8} )</td>
</tr>
<tr>
<td>fp16 (half)</td>
<td>10 bits</td>
<td>5 bits</td>
<td>( 10^{-85} ) = ( 5 \times 10^{-4} )</td>
</tr>
<tr>
<td>bfloat16 (half)</td>
<td>7 bits</td>
<td>8 bits</td>
<td>( 10^{-80} ) = ( 4 \times 10^{-3} )</td>
</tr>
</tbody>
</table>

**BLR MATRICES**

We consider a certain class of matrices, whose off-diagonal blocks have low numerical ranks. More precisely, the singular values of such blocks decrease rapidly, typically following an exponential decay. BLR compression consists in approximating each of those block as a product of two smaller rectangular matrices (low-rank approximation). It may be based on a truncated SVD or QR decomposition.

![Example of a BLR matrix (per[f009, HES psmpfun from EDF]). Color scale: numerical ranks of the blocks](image)

**LOW-RANK APPROXIMATION IN MIXED PRECISION**

We introduce a new approach to handle a low-rank approximation, in case it is based on a truncated SVD or QR decomposition. We propose to separate the columns into several groups, associated with different floating-point formats.

- A criterion for storing columns \( x_i \) and \( y_i \) in precision fp32:
  \[
  \frac{1}{\sigma_{\text{tol}}} \| B \| < \sigma_{\text{tol}} \leq \frac{1}{\epsilon} \| B \|
  \]

- Compression error:
  \[
  \| B - XY \| \leq 5\epsilon \| B \|
  \]

We plot the relative gains with mixed precision compared to double precision, as a function of the error. We notice that, for a given error, the mixed precision variant achieves better performances than the double precision (\( \times 2 \) to \( \times 3 \) in terms of storage and expected time).

*LU factorization of dense matrices*

- We emulate a LU factorization with BLR in 3 precisions: fp64, fp32 and bfloat16.
- Hypothesis: time cost = flops(fp64) + \( \frac{1}{2} \) flops(fp32) + \( \frac{1}{4} \) flops(bfloat16)
- We plot the relative gains with mixed precision compared to double precision, as a function of the error. We notice that, for a given error, the mixed precision variant achieves better performances than the double precision (\( \times 2 \) to \( \times 3 \) in terms of storage and expected time).

*Results on dense matrices*

- Our first results show that, by adding mixed precision, there is a gain of storage between \( \times 1.2 \) and \( \times 1.7 \) regarding the LU factors:

*LU factorization of sparse matrices*

A multifrontal solver, such as MUMPS[3], computes a LU factorization of a sparse matrix. In order to do that, many partial LU factorizations of smaller dense matrices are computed. The LU factors are potentially BLR matrices, and their storage cost is a large part of the memory peak. We added an option in MUMPS that converts the low-rank blocks to mixed precision when they are not used.

If mixed-precision BLR is used for storage gains only, only a conversion operation is needed for the formats. Instead of the 3 common floating-point formats, we decided to use 7, respectively on 64, 56, 48, 40, 32, 24, and 16 bits.

- Example: conversion from fp32 to "fp24":

- Hypothesis: time cost = flops(fp64) + \( \frac{1}{2} \) flops(fp32) + \( \frac{1}{4} \) flops(bfloat16)
- We emulate a LU factorization with BLR in 3 precisions: fp64, fp32 and bfloat16.
- We notice that, for a given error, the mixed precision variant achieves better performances than the double precision (\( \times 2 \) to \( \times 3 \) in terms of storage and expected time).

References

2. N.Higham, T.Mary, Solving Block Low-Rank Linear Systems by LU Factorization is Numerically Stable, IMA Journal of Numerical Analysis, 2019
3. https://mumps-solver.org

Perspectives

- Aim for times gains in MUMPS by performing computations in mixed precision
- Develop a variant of the algorithm that uses fp16 instead of bfloat16. Scaling methods will be required.
- A QR factorization algorithm may be accelerated using mixed precision