Hierarchical matrices acceleration of GMRES solver in four-dimensional Finite Element Method computations

Mateusz Dobija[1], Anna Paszyńska[1], Marcin Łoś[2], Maciej Paszyński[2]
[1] Jagiellonian University, Gołębia 24, 31-007 Kraków, Poland
[2] AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland
mateusz.dobija@doctoral.uj.edu.pl, anna.paszynska@uj.edu.pl,
los@agh.edu.pl, maciej.paszynski@agh.edu.pl

Keywords: finite element method, hierarchical matrices, GMRES, iterative solvers, space-time formulations

1. Introduction

The hierarchical matrices have been introduced by Hackbusch [1]. Usually, in scientific computing, the rank of the entire matrix is equal to the size of the matrix. This is because the low-rank matrices are not invertible [2]. However, if we partition the matrix recursively into blocks, we discover that the off-diagonal blocks are low-rank. The idea of the hierarchical matrices is to decompose the matrix recursively into off-diagonal low-rank blocks. The matrices resulting from finite element method computations decompose by recursive partitioning into the diagonal, with the low-rank off-diagonal blocks. The rows and columns in the matrices are related to basis functions spread over the computational mesh nodes. The non-zero entries in the matrices result from the overlapping of basis functions from different nodes of the mesh. The further the nodes, the less the overlap, and the lower is the rank of the off-diagonal block of the matrix.

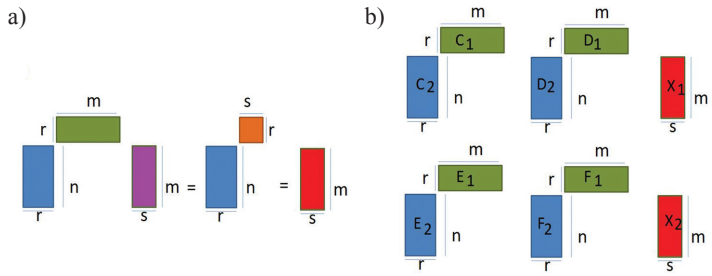

Figure 1. Panel (a) Multiplication of an (n,m)-size block of a matrix compressed with rank r by s vectors of size m. Panel (b) Multiplication of a matrix partitioned into four (n,m)-size blocks compressed with rank r by s vectors of size $2m$.

The benefit of having a matrix compressed into recursive low-rank blocks is that the multiplication of this matrix by a vector can be performed in a linear O(N) computational cost. This is illustrated in Figure 1. Panel (a) presents a single block of a matrix of size n-rows and m-columns. In the compressed format, this block is represented by a matrix of n rows and r columns, multiplied by another matrix of r rows and m columns. The low-rank blocks are kept in this compressed

The publication is co-financed from the state budget under the programme
of the Minister of Education and Science called “Excellent Science” project no. DNK/SP/548041/2022

format without multiplication back into the full block. A block can be decomposed into two blocks if it has a low-rank r. Such the block, when multiplied by s vectors of m-rows, requires O(N) floating-point operations. The first multiplication of a green block of size r-rows and m-columns by a pink vector of m-rows and s-columns is performed using O(rms) operations. The resulting orange block of r-rows and s-columns is multiplied by a blue block of n-rows and r-columns, using O(srn) operations. The total number of operations is O($mrs + nrs$) = O(max(m, n)rs) = O(N) where $N = \max(m, n)$ and $r,s << N$.

This low computational cost multiplication of blocks by multiple right-hand sides generalizes into several blocks, as presented in panel (b). This is because the right-hand side vector can be partitioned into sub-blocks of the sizes corresponding to particular sub-blocks of the matrix. The multiplication can be performed then in the following way: $C_2(C_1X_1) + D_2(D_1X_2)$ for the first block, and $E_2(E_1X_2) + F_2(F_1X_2)$ for the second block. Each of the intermediate results, namely $C_2(C_1X_1)$, and $D_2(D_1X_1)X_2)$, and $E_2(E_1X_2)$, and $F_2(F_1X_2)$ is computed in a linear computational cost, using the scheme from panel (a), and they are aggregated into the resulting vectors.

2. Four-dimensional finite element method

The four-dimensional finite element method employs discretizations over the four-dimensional computational grids. It is employed for space-time formulations. Usually, the discretization over the first three dimensions concerns the spatial dimensions, x, y, and z, and the fourth dimension concerns the time variable. The space-time formulations are widely used nowadays in the finite element method community [4–7]. The idea of the space-time formulation is that it can be applied to time-dependent problems, where the dynamic of the system is more intense in one part of the computational domain than in the others. Thus, to follow the complexity of the dynamical changes of the modeled system, we can perform smaller time steps in one part of the domain and, at the same time, larger time steps in the other parts. This can be expressed by multi-dimensional grids, where we adapt the four-dimensional hexhedrals in spatial and temporal dimensions.

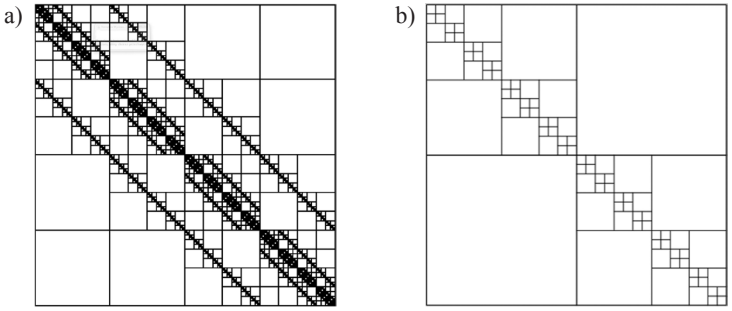

Figure 2. Panel (a) Matrix of four-dimensional finite element method compressed with rank $r = 1$ into a hierarchical matrix. Panel (b) Ideal recursive decomposition of a matrix, where we refine towards the diagonal blocks, and all off-diagonal blocks have rank 1.

As the example, on panel (a) in Figure 2, we present the structure of the hierarchical matrix (after the rank 1 compression) of the four-dimensional mesh expressing the heat transfer problem on a uniform four-dimensional mesh. In this work, we focus on developing efficient solvers for space-time formulations. We employ the hierarchical format of the matrix discretized over the four--dimensional space-time grid. We show that this matrix can be employed to speed up the iterative GMRES solver algorithm.

3. Accelerating the GMRES algorithm

The Generalised Minimum RESidual (GMRES) iterative algrithm [3] is the generalization of the Minimal RESidual (MINRES) solver. Unlike MINRES, which only works for symmetric matrices, it can be applied for unsymmetric systems. The algorithm can be summarized as follows:

1) Compute $r_0 = b - Ax_0$,
2) Compute v1 = $r_{0|} / ||r_0||$,
3) Loop $j = 1,2,\ldots,k$,
   a) Compute $h_{i,j} = (Av_j, v_i)$ **for** $i = 1,2,\ldots,j$,
   b) Compute $w_{j+1} = Av_j - \Sigma_{i,\ldots,j} h_{i,j} v_i$,
   c) Compute $h_{j+1,j} = ||w_{j+1}||_2$,
   d) Compute $v_{j+1} = w_j + 1 / h_{j+1,j}$,
4) Form solution $x_k = x_0 + V_k y_k$ where $V_k = [v_1,\ldots,x_k]$ and y_k minimizes $J(y) = ||\beta e_1 - H_k y||$ where H is the matrix of $h_{i,k}$.

In this algorithm, the matrix A is employed in line 1, where it is multiplied by the x_0 vector, as well as in lines 3a and 3b, where it is multiplied by the vector v_j. Now, with matrix A being sparse and having NNZ non-zero entries, the multiplication of a matrix by a vector can be performed with NNZ computational cost. We must multiply each non-zero entry of A by a non-zero entry of the vector. For space-time formulations, though, the NNZ can be large.

What is the cost of multiplication of the hierarchical matrix of size N by a vector? For the ideal case, presented in panel a in Figure 2, our matrix is partitioned recursively into the diagonal blocks, and the off-diagonal blocks have rank 1. The computational cost of the matrix-vector multiplication is the following. When we recursively partition the matrix into four blocks, the total cost is $C(N) = 2C(N/2) + 2O(r2N/2) + O(N)$, where $C(N)$ is the cost of multiplication for the entire matrix of size N, recursively partition into blocks into the diagonal, the $C(N/2)$ is the cost of multiplication of the recursivly partitioned into the diagonal of two halves of the original matrix, and $O(r2N/2)$ is the cost of multiplication of the off-diagonal blocks of size $N/2$ with rank r, by the vector. Additionally, we count N additions related to the aggregations of the resulting vector. The solution to this recursive equation is $C(N) = O(N\log N)$. Thus, in our method, we can perform the matrix-vector multiplications with quasi-linear computational cost.

4. Numerical experiments

To summarize the paper, we have generated the space-time matrix for the heat transfer problem over the computational mesh of size of 8 elements in each direction. In order to obtain a stable numerical solution, we employed the residual minimization method [8]. We have employed the isogeometric analysis for discretization, which can be understood as the finite element method with higher-order B-spline basis functions [9]. We have used quadratic B-splines in each direction for trial and quadratic B-splines with C0 separators for testing (equivalent to the Lagrange basis). For such the setup of the residual minimization method, the size of the matrix is ([trial space size] + [test space size])4 = ([8 + 2] + [8 + 7 + 2])4 = 274 = 531 441. Thus, we have a matrix of 531 441 rows and columns. We have employed the GMRES solver for the original uncompressed matrix, and we compared it with the matrix generated in a rank one compressed format. We set up the solver accuracy to 0.000001.

1) For the original full matrix, the GMRES solver required ten iterations with 25 112 040 total number of floating-point operations (25 112 204 flops per iteration)
2) For the compressed hierarchical matrix, the GMRES solver requires seven iterations with 2 307 760 total flops (329 680 flops per iteration).

Thus, we have shown that our method reduces the computational cost of the GMRES solver by one order of magnitude on the space-time formulations solved with isogeometric analysis methods.

References

1. Hackbush W.: *Hierarchical Matrices: Algorithms and Analysis*. Springer, 2009.
2. Horn R.A., Horn R.A., Johnson C.R.: *Matrix analysis*. Cambridge university press, 1990.
3. Saad Y.: *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2nd edition, 2003.
4. Schafelner A., Vassilevski P.S.: *Numerical Results for Adaptive (Negative Norm) Constrained First Order System Least Squares Formulations*. Computers and Mathematics with Applications, 95, 2020, 256–270.
5. Voronin K., Lee C.S., Neumüller N., Sepulveda-Salaz P., Vassilevski P.S.: *Space-time discretizations using constrained first-order system least squares (CFOSLS)*. Journal of Computational Physics, 373, 2018, 863–876.
6. Steinbach O., Yang H.: *Comparison of algebraic multigrid methods for an adaptive space–time finite--element discretization of the heat equation in 3D and 4D*. Numerical Linear Algebra with Applications, 2018.
7. Demkowicz L., Gopalakrishnan J., Nagaraj S., Sepúlveda P.: *A Spacetime DPG Method for the Schrödinger Equation*. SIAM Journal of Numerical Analysis, 2016.
8. Chan J., Evans J.A.: *A Minimum-Residual Finite Element Method for the Convection-Diffusion Equations*. ICES-Report 13–12, 2013.
9. Cottrell J.A., Hughes T.J.R., Bazilevs Y.: *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009.

Acknowledgements. This work is partially supported by The European Union's Horizon 2020 Research and Innovation Program of the Marie Skłodowska-Curie grant agreement No. 777778, MATHROCKs. Scientific paper published within the framework of an international project co--financed with funds from the program of the Ministry of Science and Higher Education entitled „PMW" in years 2022-2023; contract no. 5243/H2020/2022/2.