ParCFD'2021
32^{nd} International Conference on Parallel Computational Fluid Dynamics
May-17-19 2021, Nice FRANCE

ALGEBRAIC LINELET PRECONDITIONER FOR THE SOLUTION OF THE POISSON EQUATION ON BOUNDARY LAYER FLOWS.

Ramiro de Olazábal†, Oriol Lehmkuhl, Guillaume Houzeaux and Ricard Borrell

†Barcelona Supercomputing Center
Department of Computer Applications in Science and Engineering
08024, Barcelona, Spain
ramiro.deolazabal@bsc.es

Key words: Preconditioned Conjugate Gradient, Krylov Methods, Parallel Algebraic Linelet Preconditioner, Boundary Layer Flow

Abstract. An algebraic linelet preconditioner is presented, which works in parallel regardless of the mesh's geometry and without imposing constraints on the domain partition. It is designed to deal with highly anisotropic meshes. A key aspect of this work is developing an algorithm that generates the preconditioning matrix by purely algebraic considerations. This preconditioned is coupled to Alya, the in-house HPC multi-physics code developed at Barcelona Supercomputing Center.

1 INTRODUCTION

Navier-stokes equations for incompressible flows can be solved using the fractional step projection method, by which the pressure solution is decoupled from the rest of the equations. In this context, a Poisson's equation for the pressure correction equation needs to be solved at least once per time-step on this scheme [2]. This step represents the primary source of performance bottlenecks of the code and is one of the most time-consuming and difficult to parallelize. Furthermore, when problems involving boundary layer flow need to be simulated, highly anisotropic meshes are employed to accurately describe this critical region. Such compression of the mesh degrades the conditioning of the linear system associated with the Poisson equation. This makes the solver even more expensive in terms of time and computational resources employed. Figure 1 shows an example of a mesh for the numerical simulation of an airplane wing; here, the nodes located in the prismatic boundary layer represent the 46.7% of the total nodes. Thus its impact on Poisson's equation discretization is not negligible.

Motivated by these facts, we developed a parallel preconditioner able to boost the performance of the Preconditioned Conjugate Gradient (PCG) solver for meshes with high anisotropy in the boundary layer. In particular, we aimed to extend the capabilities of the linelet preconditioner [5]. This kind of preconditioner considers only the two strongest couplings for each node of the boundary layer. As a result, the system is decomposed

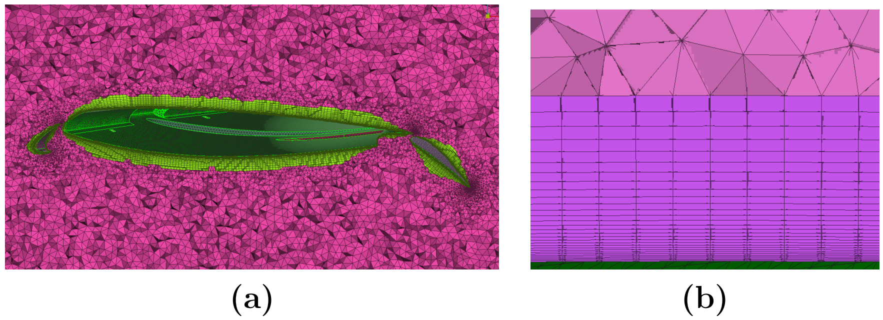

Figure 1: Example of a mesh for the numerical simulation of an airplane wing (Figure a) and zoom of the same mesh showing the prismatic boundary layer (Figure b).

into a set of one-dimensional tridiagonal subsystems. A fully algebraic approach independent of the mesh partitioning is presented, which works for Finite Elements (FE), Finite Differences (FD), and Finite Volumes (FV) methods. The authors are not aware of publications with linelet preconditioner implementations of similar capabilities.

2 Algorithm

Linelets are assembled, choosing the highest couplings between nodes. Algebraically, the linear system matrix A is approximated by a given matrix M, which consists of the same diagonal entries as A and some of the non-diagonal ones. Hence building matrix M can be thought of as filtering matrix A.

The main idea of the algorithm is as follows. Suppose we start with an initial symmetric matrix A as in Figure 2a. From each row, we select either one or no element Aij from the non-diagonal entries located at one side of the main diagonal, for example, $Ai, j > i$ (Figure 2b). One important constraint to this selection is that from every row and every column, either none or one element is chosen, i.e., there can not be two or more selected entries occupying the same row or column. Finally, every non-diagonal coupling $A_{ij,j\neq i}$ that has not been selected is filtered and considered the selected elements and their symmetric counterparts. We then get Figure 2c. Note that each node will be coupled at most with two other nodes (Figure 2d).

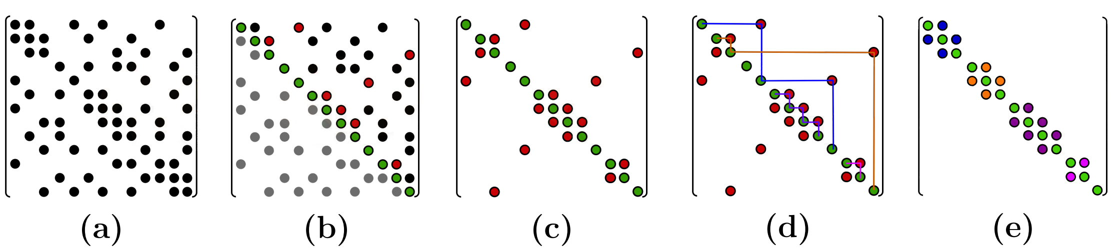

Figure 2: Algorithm for building a linelet preconditioning matrix algebraically: a) Input matrix, b) Filtering matrix, c) Filtered matrix, d) Linelet structure.

The second step is to solve the system $Ms = r$ for s, for which the TDMA [3] is combined with a Dual Schur Decomposition Method [4], in order to deal with the situation of linelets traversing various subdomains. It is important to note that this operation requires just one communication step to replicate the total interface values in each subdomain.

3 Preliminary Results

Alya[1], our in-house HPC multi-physics code developed in our research group, already has an in-built linelet preconditioner to deal with highly anisotropic meshes. Nevertheless, it works by assembling the linelets within each subdomain without allowing communications between them.
A Preconditioned Conjugate Gradient (PCG) solver was developed, which allows the linelets to spread over an arbitrary number of processes. Up to this moment, this solver works for structured meshes and builds the linelets by geometric consideration. However, we aim to extend it to non-structured meshes and to implement the algorithm described in 2 to build the preconditioning matrix algebraically.

Comparing both cases mentioned above, it was studied how cutting the linelets affects the convergence of PCG. For this to be done, the global domain was partitioned in the z direction. Each subdomain had its own set of linelets, disconnected from the rest. An example was run for a cubic mesh whose dimensions were $160 \times 240 \times\ 128$ (expressed in nodes, the number of nodes in each direction will be represented by N_x, N_y and N_z). The mesh is refined in the z-direction, being the mesh highly compressed near $z = 0$ and with refinement function

$$\Delta z_i = 2H - H\left\{1 + \frac{1}{tanh(\rho)} tanh\left[\rho(1 - \frac{i+i_0}{N_z})\right]\right\}, \text{i=1,...,}N_z. \quad (1)$$

Figure 3 shows the number of iterations needed for a different amount of partitions.

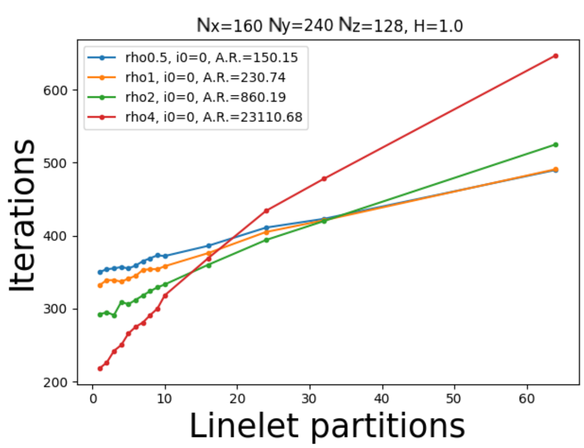

Figure 3: Number of iterations required to achieve a residual lower than 10^{-6} for different number of partitions in the z direction, $\rho = 1$.

It can be seen from Figure 3 that if linelets are spread over many processes but there is no communication between its different sections, the solver needs to perform more iterations to reach the same tolerance. This result justifies the use of a direct parallel solver like the Dual Schur Decomposition Method. For all the cases shown in Figure 3, it was asserted that when the Schur algorithm is implemented into the solver, both the final residual and the difference with the exact answer were the same as in the no partitions case (linelet partitions equal to 0 in Figure 3).

4 Conclusions

To sum up, in this work will it will be developed an algebraic linelet preconditioner, together with all the computational and algorithmic tools required for it to work in parallel regardless of the geometry of the problem and the domain partition. This preconditioner will then be coupled to Alya, the in-house HPC multi-physics code developed in our research group. In the final work, the solver will be assessed from the scalability and robustness point of view using meshes similar to those shown in Figure 1, coming from relevant for aeronautical applications.

REFERENCES

[1] https://www.bsc.es/research-development/research-areas/engineering-simulations/alya-high-performance-computational. Accessed: 14-01-2021.

[2] J. Perot. An analysis of the fractional step method. *Journal of Computational Physics*, 108:51–58, 09 1993.

[3] R. S. F. Quarteroni, Alfio; Sacco. *Numerical Mathematics*. 2007.

[4] M. Soria, C. Pérez-Segarra, K. Claramunt, and C. Lifante. A direct algorithm for the efficient solution of the poisson equations arising in incompressible flow problems. In P. Wilders, A. Ecer, N. Satofuka, J. Periaux, and P. Fox, editors, *Parallel Computational Fluid Dynamics 2001*, pages 331 – 338. North-Holland, Amsterdam, 2002.

[5] O. Soto, R. Lohner, and F. Camelli. A linelet preconditioner for incompressible flows. *International Journal of Numerical Methods for Heat Fluid Flow*, 13:133–147, 2003.