Computer Methods in Applied Mechanics and Engineering 431 (2024) 117249

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Efficient AMG reduction-based preconditioners for structural mechanics

Àdel Alsalti-Baldellou [a,c,*], Andrea Franceschini [b], Gianluca Mazzucco [b], Carlo Janna [b,c]

[a] Department of Information Engineering, University of Padova, Via Giovanni Gradenigo, 6b, 35131 Padova PD, Italy
[b] Department of Civil, Environmental and Architectural Engineering, University of Padova, Via Francesco Marzolo, 9, 35131 Padova PD, Italy
[c] M³E S.r.l., Via Giambellino, 7, 35129 Padova PD, Italy

ARTICLE INFO

MSC:
65F08
65N55
65Y05

Keywords:
Periodic structures
Spatial symmetries
Non-linear problems
Algebraic multigrid
Multigrid reduction

ABSTRACT

Structural problems play a critical role in many areas of science and engineering. Their efficient and accurate solution is essential for designing and optimising civil engineering, aerospace, and materials science applications, to name a few. When appropriately tuned, Algebraic Multigrid (AMG) methods exhibit a convergence that is independent of the problem size, making them the preferred option for solving structural problems. Nevertheless, AMG faces several computational challenges, including its remarkable memory footprint, costly setup, and the relatively low arithmetic intensity of the sparse linear algebra operations. This work presents AMGR, an enhanced variant of AMG that mitigates such limitations. Its name arises from the AMG reduction framework it introduces, and its flexibility allows for leveraging several features that are common in structural problems. Namely, periodicities, spatial symmetries, and localised non-linearities. For such cases, we show how to reduce the memory footprint and setup costs of the standard AMG, as well as increase its arithmetic intensity. Despite being lighter than the standard AMG, AMGR exhibits comparable scalability and convergence rates. Numerical experiments on several industrial applications prove AMGR's effectiveness, resulting in up to 3.7x overall speed-ups compared to the standard AMG.

1. Introduction

Civil engineering design often relies on simplified numerical models, such as 1D or 2D discretisations, to quickly assess overall information like internal loading or global kinematic quantities. This approach is particularly useful for linear analyses or when only small non-linearities are present. Frame and shell elements can be effectively used in the design process if the actual structure can be reasonably approximated under the strict assumptions these elements rely on [1–4]. However, cases involving a significant non-linearity require a more precise geometric description and detailed material characterisation. Typically, a three-dimensional approach is employed to evaluate the failure or dissipative mechanisms exhibited by a civil structure under extreme loading conditions such as earthquakes, explosions and fires [5–7], as well as to study historic structures [8]. Developing 3D civil structural models is time-consuming and requires substantial hardware resources to obtain the analysis solution. Therefore, optimised numerical procedures are essential to enhance the usability and efficiency of this type of analysis.

Structural analysis is also vital in designing and optimising automotive components, like the chassis, suspension systems, and crash structures of the vehicles [9]. Similarly, it is essential in the aircraft industry to predict fatigue and identify possible failure

* Corresponding author at: Department of Information Engineering, University of Padova, Via Giovanni Gradenigo, 6b, 35131 Padova PD, Italy.
E-mail address: adel.alsaltibaldellou@unipd.it (A. Alsalti-Baldellou).

https://doi.org/10.1016/j.cma.2024.117249
Received 5 June 2024; Received in revised form 15 July 2024; Accepted 19 July 2024
Available online 7 August 2024

modes of the various components, such as the wings and fuselages [10,11]. Other applications include the study of biomechanical processes like bone or cartilage deformation under different loading conditions [12,13]; the simulation of manufacturing techniques like stamping, forging and casting for predicting the performance and durability of manufactured goods [14,15]; and in offshore engineering, the assessment of the stability of the structures in response to wave loads [16].

It is clear, then, how ubiquitous structural problems are, and how crucial their efficient solution results. Since the beginning of numerical analysis, iterative methods based on Krylov subspaces have been employed to tackle their resolution. As discussed in [17–19], they are simple to implement and easily parallelisable, solely relying on basic linear algebra kernels like the matrix by vector product, the linear combination and the dot product of vectors. Nevertheless, iterative linear solvers are very sensitive to the characteristics of the problem at hand [20–22], and to be effective, they generally require robust preconditioners [23,24], whose selection, design, and implementation are far from trivial and remain very active research areas.

In the early years of numerical linear algebra, incomplete factorisations were very popular, as evidenced by works such as [25,26]. In those pioneering articles, factorisation occurred without fill-in, leading to a Cholesky factor with a pattern equal to that of the lower part of A. Subsequent advancements introduced strategies like dynamic fill-in control [27,28]. However, the development of parallel computing in the late 1990s made alternatives with greater parallelism overshadow the traditional (and inherently sequential) methods based on incomplete factorisations. One such alternative is the preconditioners based on approximate inverses. Their construction is fairly concurrent, and, most notably, their application relies on the `SpMV`, a trivially parallelisable kernel. Popular examples include AINV [29–31], SPAI [32], and the Factored Sparse Approximate Inverse (FSAI) [33,34].

In the current landscape of solving extreme-scale linear systems on massively parallel supercomputers, the issue of scalability has become increasingly critical [35,36]. Despite offering great parallelism, single-level approximate inverses are not optimal, as the number of iterations required to meet a prescribed accuracy grows with the problem size. To address this challenge, multilevel preconditioners like Geometric Multigrid (GMG) [37,38] or Algebraic Multigrid (AMG) [39–41] emerged as effective solutions. By leveraging a combination of smoother and coarse-grid correction, multigrid methods often solve PDEs at a convergence rate that is independent of the problem size. Numerous freely available packages offering high-quality and very scalable implementations exist. See, for instance, Hypre [42], Trilinos [43], and PETSc [44] libraries.

In order to improve the performance of the linear solvers, it is imperative to identify and address the challenges posed by the current computing devices, developing novel strategies that overcome them. To name a few, introducing mixed precision [45] or adapting more compute-intensive algorithms [46] effectively mitigates the low arithmetic intensity of the sparse linear algebra kernels. Similarly, the increasing ratio of the network to the memory bandwidths spurred the creation of implementations that either conceal or completely eliminate inter-node communications [47–49]. Additionally, the limited available memory motivated strategies like exploiting data sparsity [50–52]. In the recent years, all such improvements have gained paramount importance in the context of data-driven modelling [53,54], where there is an urgency to produce vast quantities of training and testing data from traditional numerical simulations.

Several studies leveraging spatial symmetries for solving Poisson's equation exist [55–57]. This work extends them by presenting AMGR, an enhanced variant of AMG whose name arises from the AMG reduction framework on which it is based. Multigrid reduction has been successfully applied in different contexts in the recent years [58,59]. Despite being initially defined in the context of computational fluid dynamics (CFD) simulations [60], the present work leverages the flexibility of AMGR to make it take advantage of several features that are common in structural problems. Namely, periodicities, spatial symmetries, and localised non-linearities. In such cases, AMGR has multiple benefits. On the one hand, it introduces an aggressive coarsening to the top level of the multigrid hierarchy that, despite making AMGR significantly lighter, preserves the excellent convergence of the standard AMG. On the other, it makes the application of the top-level smoother faster and more efficient thanks to replacing the standard `SpMV` with the more compute-intensive `SpMM` [61,62]. As a result, AMGR is lighter and more compute-intensive than the standard AMG, converges and scales equally well, and yields up to 3.7x overall speed-ups.

Thanks to its flexibility, the range of industrial applications compatible with AMGR is considerably extensive. For instance, translational and rotational symmetries are commonly used to simplify the analysis of complex structures by breaking them down into smaller repeating units [63–65], such as the bricks forming a wall or the symmetric part of a bridge pier. Similarly, reflection symmetries arise in a wide variety of industrial applications, such as wheel rim failure, helical gear optimisation and bearing vibrations [66,67]. Furthermore, AMGR accelerates the non-linear analysis of problems in which the non-linearity is limited to a smaller region within the domain, as happens when simulating the failure response of steel joints [68], modelling the contact between continuous domains [69] or simulating elasto-plastic deformation within a reservoir [70,71], to name a few. In all such problems, AMGR greatly reduces the setup costs by reusing the (large) portions of the preconditioner that remain constant between consecutive non-linear iterations.

The remaining sections of the paper are organised as follows. Section 2 reviews the main components of AMG preconditioners. Section 3 defines the novel AMGR preconditioner for structural problems, detailing its application in different scenarios. Section 4 discusses the performance of AMGR on industrial problems, and finally, Section 5 gives some concluding remarks.

2. Overview of algebraic multigrid

For the sake of completeness, let us start by reviewing the main characteristics of Algebraic Multigrid (AMG). It is a class of powerful methods for the efficient solution of linear systems of equations of the form:

$$Ax = b, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ and $x, b \in \mathbb{R}^n$.

AMG can be used as a standalone solver or as a preconditioner to accelerate the convergence of Krylov-subspace methods like PCG [72], GMRES [73] or BICGSTAB [74]. Even though there are multiple implementations able to tackle general nonsymmetric or saddle-point problems, AMG was originally designed for symmetric positive-definite (SPD) matrices, on which it performs best. Thus, we will restrict our attention to this context, assuming that A is always SPD. Regarding the choice of AMG, it is mainly motivated by its optimality, that is, by its capability of providing convergence rates independent (or mildly dependent) of the grid size. This feature makes AMG particularly attractive for large-scale problems, as it guarantees solution times growing linearly with the grid size. It reduces the error, $e_k = A^{-1}b - x_k$, through a multilevel approach that effectively removes components belonging to both the low- and high-frequency parts of A's eigenspectrum. When appropriately tuned, AMG allows for a convergence rate which does not depend on the mesh size, allowing the resolution of linear systems with millions, or even billions, of unknowns in reasonable timeframes.

For simplicity, let us consider a two-level AMG method. Its application starts by applying a basic iterative method, like (damped) Jacobi or (block) Gauss–Seidel, to the linear system of (1):

$$x_{k+1/2} = (\mathbb{I}_n - M^{-1}A)x_k + M^{-1}b, \tag{2}$$

where $\mathbb{I}_n \in \mathbb{R}^{n\times n}$ denotes the identity matrix of order n and M^{-1} the smoother, that is, a preconditioner $M \in \mathbb{R}^{n\times n}$ such that $M^{-1} \simeq A^{-1}$. In the case of Jacobi, for instance, $M = \mathrm{diag}(A)$. This process, termed relaxation or smoothing, is repeated a small number of times, ν_1, due to the rapid deterioration of its convergence. Such deterioration is a well-known drawback of basic iterative methods, which rapidly dampen the high-frequency error components but are ineffective in removing the low-frequency ones. To remedy this, AMG reduces the low frequencies by treating them as the high frequencies of A_c, a coarse representation of A obtained by projecting A onto a coarser space:

$$A_c = P^T A P \in \mathbb{R}^{n_c\times n_c}, \tag{3}$$

in which $P \in \mathbb{R}^{n\times n_c}$ is the prolongation operator responsible for transferring the information between the coarse and fine levels.

The construction of A_c is crucial for obtaining effective AMG methods. This work adopts a Classical AMG approach, in which the coarse level unknowns are a subset of those in the fine level. Such a subset is selected as follows:

1. The edges between the nodes of the adjacency graph of A are marked as either weak or strong, giving rise to the so-called Strength of Connection (SoC) graph. Strong connections represent the directions along which the smooth error varies slowly and, classically, include those edges satisfying:

$$-a_{ij} > \theta \max_{k\neq i}\{a_{ik}\}, \tag{4}$$

   where $\theta \in \mathbb{R}$ is a threshold parameter typically set to around 0.25.
2. After setting the SoC graph, the subset of coarse nodes is formed by applying a Maximum Independent Set (MIS) algorithm. The subset of coarse nodes must be independent, to avoid bringing redundant information to the coarse level, and maximal, to cover the original SoC graph entirely.

Then, the smoothed residual, $r_{k+1/2} = b - Ax_{k+1/2}$, is projected onto the coarse space, where a correction vector is computed by inverting A_c:

$$h_c = A_c^{-1} P^T r_{k+1/2}. \tag{5}$$

Subsequently, h_c is interpolated back to the fine space, $h = Ph_c$, and applied to $x_{k+1/2}$. Typically, another relaxation step is applied to the corrected solution to obtain the final approximation:

$$x_{k+1} = (\mathbb{I}_n - M^{-T}A)(x_{k+1} + h) + M^{-T}b. \tag{6}$$

Remarkably enough, the post-smoothing step makes use of M's transpose in order to make the overall AMG operator symmetric, regardless of M. By doing so, it can be used as a preconditioner for PCG.

In a two-grid scheme, the error propagation operator, E_{TG}, defines the combined action of the relaxation and the coarse-grid correction:

$$E_{TG} = (\mathbb{I}_n - M^{-T}A)^{\nu_2}(\mathbb{I}_{n_c} - PA_c^{-1}P^TA)(\mathbb{I}_n - M^{-1}A)^{\nu_1} \in \mathbb{R}^{n\times n}, \tag{7}$$

where ν_1 and ν_2 denote the number of smoother sweeps in the pre- and post-relaxation steps, respectively, and, to ensure the preconditioner's symmetry, we have that $\nu_1 = \nu_2$. A rapidly converging AMG exhibits an error propagator of norm close to zero independently of the grid size. This condition generally holds when the smoother and coarse-grid correction complement each other well, meaning that the error components not damped by the relaxation, *i.e.*, the algebraically smooth components, are effectively removed by the coarse-grid correction.

The coarsening ratio, *i.e.*, the ratio of n_c to n, is generally of the order of 10^{-1}. Hence, for realistically sized problems, the costs of inverting A_c are exceedingly large. To overcome this limitation, AMG approximates the action of A_c^{-1} by nesting another two-grid method. In practise, this is done recursively until the size of the coarsest level allows for a direct solution. Although analogous schemes exist, AMG is often applied as a V-cycle, that is, by moving from the finest to the coarsest level and back to the finest.

Algorithm 2.1 Recursive AMG Setup

procedure AMG_SetUp(A_k)
  Define Ω_k as the set of the n_k vertices of the adjacency graph of A_k;
  if n_k is small enough to allow a direct factorisation **then**
    Compute $A_k = L_k L_k^T$;
  else
    Compute M_k such that $M_k^{-1} \simeq A_k^{-1}$;
    Define the smoother as $S_k = \left(\mathbb{I}_{n_k} - M_k^{-1} A_k\right)$;
    Partition Ω_k into the two disjoint sets C_k and F_k via coarsening;
    Compute the prolongation matrix P_k from C_k to Ω_k;
    Compute the new coarse level matrix $A_{k+1} = P_k^T A_k P_k$;
    AMG_SetUp$\left(A_{k+1}\right)$;
  end if
end procedure

Algorithm 2.2 AMG application in a V(ν_1, ν_2)-cycle

procedure AMG_Apply(A_k, y_k, z_k)
  if k is the last level **then**
    Forward and Backward solve $L_k L_k^T z_k = y_k$;
  else
    Smooth ν_1 times $A_k s_k = y_k$ starting from 0;
    Compute the residual $r_k = y_k - A_k s_k$;
    Restrict the residual to the coarse grid $r_{k+1} = P_k^T r_k$;
    AMG_Apply$\left(A_{k+1}, r_{k+1}, h_{k+1}\right)$;
    Prolongate the correction to the fine grid $h_k = P_k h_{k+1}$;
    Update $s_k \leftarrow s_k + h_k$;
    Smooth ν_2 times $A_k z_k = y_k$ starting from s_k;
  end if
end procedure

The basic steps required to compute and apply a generic AMG are schematically presented in Algorithms 2.1 and 2.2, respectively. The building blocks of any AMG implementation are the smoother, the coarsening strategy and the prolongation operator. In this work, we use the AMG provided by the Chronos library [75,76]. It is a classical AMG, meaning that the coarsening algorithm classifies the unknowns into fine and coarse and uses the latter as the primary unknowns for the coarser level. Conversely, in aggregation-based AMG, the coarsening is obtained by aggregating the unknowns. It is usually preferred over classical AMG for solving structural problems, as it is easier to accommodate the multidimensional near-kernel subspace represented by the rigid body modes. Nevertheless, the Dynamic Pattern Least Squares (DPLS) prolongation employed in Chronos allows for an accurate and sparse representation of multidimensional subspaces within a classical AMG framework [77]. Another key component of Chronos is the FSAI smoother [33,78], which is extremely helpful for solving ill-conditioned problems. In the remainder of this paper, we will show how to greatly improve these three components on problems with periodicities, spatial symmetries, or localised non-linearities.

3. AMG reduction framework

This section presents AMGR, an AMG reduction framework previously introduced in the context of CFD simulations [60]. We will define it in general terms and then detail its extension towards structural problems. Namely, towards periodic structures, domains with symmetries, and problems with a non-linearity only applying to a subset of unknowns.

AMGR arises from applying a non-overlapping domain decomposition that, as Sections 3.1–3.3 will detail, must be consistent with the problem at hand to exploit its advantages. Fig. 1 sketches a domain divided into n_b subdomains and the resulting classification of the unknowns, which splits them into inner ($\Omega_1 \cup \ldots \cup \Omega_{n_b}$) and interface ($\Gamma_1 \cup \ldots \cup \Gamma_{n_b}$) unknowns.

Regardless of the decomposition, the number of inner unknowns should be much larger than the number of interface unknowns, *i.e.*, $n_{\text{inn}} \gg n_{\text{ifc}}$. Additionally, thanks to reordering the coefficient matrix by first indexing the inner and then the interface unknowns, it satisfies the following block structure:

$$A = \begin{pmatrix} \bar{K} & \bar{B} \\ \bar{B}^T & \bar{C} \end{pmatrix} \in \mathbb{R}^{n \times n}, \tag{8}$$

where $\bar{K} \in \mathbb{R}^{n_{\text{inn}} \times n_{\text{inn}}}$, $\bar{B} \in \mathbb{R}^{n_{\text{inn}} \times n_{\text{ifc}}}$, $\bar{C} \in \mathbb{R}^{n_{\text{ifc}} \times n_{\text{ifc}}}$.

Even if we have not yet detailed how to define adequate decompositions, the structure of Eq. (8) suggests defining an AMG reduction by (only) setting the interface unknowns as coarse. Unfortunately, this results in exceedingly large fine-coarse distances and

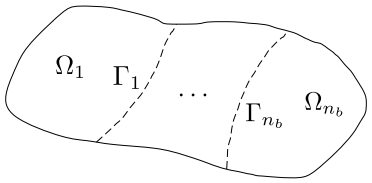


Fig. 1. Schematic representation of a non-overlapping domain decomposition. Dashed lines identify the interface.

to allow for an accurate prolongation, we need to convert some inner nodes into coarse. Algorithm 3.1 summarises the population algorithm used to ensure a maximum interpolation distance k.

Algorithm 3.1 Interface population at a maximum interpolation distance k

1: **procedure** POPULATE_COARSENING(A, k)
2: Measure the strength-of-connection of A
3: Filter the resulting adjacency graph, T
4: Compute symbolic power of T, T^k
5: Compute a maximum independent set on T^k, MIS(T^k)
6: Populate coarsening with MIS(T^k)
7: **end procedure**

For the sake of simplicity, from this point, the inner and interface will not refer to the original decomposition (whichever it is) but to their populated counterparts. Likewise, n_{ifc} will designate the enlarged interface and n_{inn} the remaining inner unknowns. Then, we can introduce the following prolongation:

$$P := \begin{pmatrix} \bar{W} \\ \mathbb{I}_{n_{\text{ifc}}} \end{pmatrix} \in \mathbb{R}^{n \times n_{\text{ifc}}}, \tag{9}$$

where $\bar{W} \in \mathbb{R}^{n_{\text{inn}} \times n_{\text{ifc}}}$ and $\mathbb{I}_{n_{\text{ifc}}} \in \mathbb{R}^{n_{\text{ifc}} \times n_{\text{ifc}}}$. The larger k is, the more effective the AMG reduction becomes in reducing the problem size. However, this comes at the price of making P inaccurate. As discussed in Section 4, using $k = 2$ provides an optimal balance.

The remaining pieces of the AMG reduction framework follow naturally. Firstly, the top-level smoother is defined as:

$$M^{-1} := \begin{pmatrix} \bar{M}_K^{-1} & \\ & \bar{M}_C^{-1} \end{pmatrix} \in \mathbb{R}^{n \times n}, \tag{10}$$

where $\bar{M}_K^{-1} \simeq \bar{K}^{-1}$ and $\bar{M}_C^{-1} \simeq \bar{C}^{-1}$. In Eq. (9), we can ignore A's off-diagonal blocks without harming the overall quality of AMGR. Indeed, overly accurate smoothers are ineffective in accelerating convergence and generally lead to a slower solution, as they do not yield a reduction in the iterations to offset its more expensive setup and application. In fact, the numerical experiments showed that thanks to being $\bar{B}$ and $\bar{B}^T$ substantially sparser than $\bar{K}$ and $\bar{C}$, we can safely ignore them for relaxation purposes, causing no significant convergence degradation and, as will be discussed in the following sections, leaving room for optimisations in the construction and application of the top-level smoother.

On the other hand, the reduced operator, $A_c \in \mathbb{R}^{n_{\text{ifc}} \times n_{\text{ifc}}}$, reads:

$$A_c := P^T \begin{pmatrix} \bar{K} & \bar{B} \\ \bar{B}^T & \bar{C} \end{pmatrix} P = \bar{W}^T \bar{K} \bar{W} + \bar{W}^T \bar{B} + \bar{B}^T \bar{W} + \bar{C}, \tag{11}$$

whose inverse is approximated through a standard AMG, $M_c^{-1} \simeq A_c^{-1}$, leading to the following error propagation operator:

$$E_{\text{AMGR}} = (\mathbb{I}_n - M^{-T})^{\nu_2} (\mathbb{I}_n - P M_c^{-1} P^T)(\mathbb{I}_n - M^{-1})^{\nu_1}, \tag{12}$$

where ν_1 and ν_2 denote the number of pre- and post-smoothing sweeps, respectively.

Let us note that, regardless of the domain decomposition, AMGR introduces an aggressive coarsening that makes the cost of the two-grid correction relatively small. Additionally, imposing an adequate inner-interface partitioning allows for further numerical and computational advantages. Namely, for a reduction of the memory footprint and setup costs of the top-level smoother, and for an increase of the arithmetic intensity relative to a standard AMG. The following subsections will discuss the selection of appropriate decompositions fully exploiting AMGR's advantages.

3.1. Substructuring

The most straightforward application of AMGR is to structural problems whose geometry arises from repeating a "substructure", such as the bricks forming a wall or the slices of a pillar. Then, by assigning a different subdomain to each substructure analogously

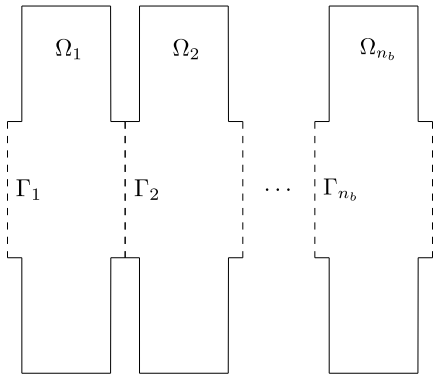


Fig. 2. Adequate domain decomposition of periodic structures. Dashed lines identify the interface.

to Fig. 2, we obtain a coefficient matrix satisfying Eq. (8). Additionally, if we apply the same local ordering to each subdomain, we ensure the following property:

$$\bar{K} = \mathbb{I}_{n_b} \otimes K \text{ and } \bar{M}_K^{-1} = \mathbb{I}_{n_b} \otimes M_K^{-1}, \tag{13}$$

where $K \in \mathbb{R}^{n_{\text{inn}}/n_b \times n_{\text{inn}}/n_b}$ stands for the restriction of $\bar{K}$ to the substructure, and $M_K^{-1} \simeq K^{-1}$ (see Fig. 2).

Eq. (13) is very convenient as it allows for assembling K and M_K^{-1} instead of the entire $\bar{K}$ and $\bar{M}_K^{-1}$. Consequently, AMGR reduces the setup costs and memory footprint of $\bar{M}_K^{-1}$ by a factor of n_b, which is particularly positive given the resources required by the top-level smoother, M^{-1}, within a multigrid hierarchy. Furthermore, the decomposition of Eq. (13) allows the standard SpMV by $\bar{K}$ to be replaced with the faster SpMM, which algebraically corresponds to:

$$\texttt{SpMV}: \begin{pmatrix} y_1 \\ \vdots \\ y_{n_b} \end{pmatrix} = \begin{pmatrix} K & & \\ & \ddots & \\ & & K \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_{n_b} \end{pmatrix} \in \mathbb{R}^{n_{\text{inn}}} \implies \texttt{SpMM}: (y_1 \dots y_{n_b}) = K(x_1 \dots x_{n_b}) \in \mathbb{R}^{n_{\text{inn}}/n_b \times n_b}. \tag{14}$$

The fact that SpMM reads K n_b fewer times makes its arithmetic intensity considerably higher, and since SpMV and SpMM are generally memory-bound kernels, this increase translates into significant speed-ups. Notably, SpMM can also be leveraged on $\bar{M}_K^{-1}$ when using smoothers that are applied through SpMV, such as FSAI.

3.2. Reflection symmetries in structural problems

AMGR can also be applied to structural problems presenting reflection symmetries. However, in such cases, the mixed components of stress and strain change sign when a reflection is applied. For example, ε_{xy} becomes $-\varepsilon_{xy}$ after a reflection along the x-orthogonal plane. Thus, the stiffness matrix of a body does not coincide with that of its reflections.

To better understand the relation between them, let us consider a degree of freedom-based ordering, in which we first index all the x, then all the y and finally all the z degrees of freedom. Then, the stiffness matrix reads:

$$A = \begin{pmatrix} A_{xx} & A_{xy} & A_{xz} \\ A_{yx} & A_{yy} & A_{yz} \\ A_{zx} & A_{zy} & A_{zz} \end{pmatrix} \in \mathbb{R}^{n \times n}, \tag{15}$$

whereas the stiffness matrix of its mirroring (along the x-orthogonal plane) reads:

$$A'_x = \begin{pmatrix} A_{xx} & -A_{xy} & -A_{xz} \\ -A_{yx} & A_{yy} & A_{yz} \\ -A_{zx} & A_{zy} & A_{zz} \end{pmatrix} \in \mathbb{R}^{n \times n}. \tag{16}$$

Remarkably enough, both can be related through the following reflection matrix:

$$R_x = \begin{pmatrix} -\mathbb{I}_{n/3} & & \\ & \mathbb{I}_{n/3} & \\ & & \mathbb{I}_{n/3} \end{pmatrix} \in \mathbb{R}^{n \times n}, \tag{17}$$

which satisfies that $R_x^{-1} = R_x$, and transforms A into A'_x:

$$A'_x = R_x A R_x. \tag{18}$$

To extend AMGR's advantages towards structural problems with symmetries, we need to take into account the considerations above. For simplicity, let us consider a body analogous to Fig. 3, *i.e.*, with a single symmetry along the x-orthogonal plane. Then,

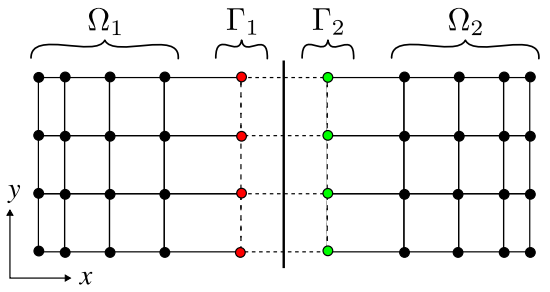


Fig. 3. Arbitrary domain with a single symmetry identifying the inner, $\Omega_1 \cup \Omega_2$, and interface nodes, $\Gamma_1 \cup \Gamma_2$.

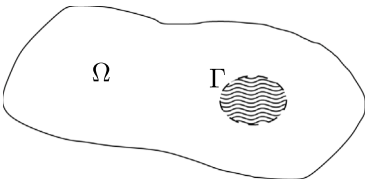


Fig. 4. Adequate domain decomposition for non-linear analysis. The wavy pattern identifies the interface.

the global stiffness matrix reads:

$$A = \begin{pmatrix} A_{\Omega_1\Omega_1} & & A_{\Omega_1\Gamma_1} & \\ & A_{\Omega_2\Omega_2} & & A_{\Omega_2\Gamma_2} \\ A_{\Gamma_1\Omega_1} & & A_{\Gamma_1\Gamma_1} & A_{\Gamma_1\Gamma_2} \\ & A_{\Gamma_2\Omega_2} & A_{\Gamma_2\Gamma_1} & A_{\Gamma_2\Gamma_2} \end{pmatrix} \in \mathbb{R}^{n\times n}. \tag{19}$$

Recalling the notation of Eq. (8), let us observe that:

$$\bar{K} = \begin{pmatrix} A_{\Omega_1\Omega_1} & \\ & A_{\Omega_2\Omega_2} \end{pmatrix}, \quad \bar{B} = \begin{pmatrix} A_{\Omega_1\Gamma_1} & \\ & A_{\Omega_2\Gamma_2} \end{pmatrix}, \quad \bar{C} = \begin{pmatrix} A_{\Gamma_1\Gamma_1} & A_{\Gamma_1\Gamma_2} \\ A_{\Gamma_2\Gamma_1} & A_{\Gamma_1\Gamma_1} \end{pmatrix}, \tag{20}$$

and, similarly to Eq. (13), by applying the same local ordering to each subdomain and denoting $K \equiv A_{\Omega_1\Omega_1}$, we have that:

$$\bar{K} = \begin{pmatrix} K & \\ & R_\Omega K R_\Omega \end{pmatrix} \text{ and } \bar{M}_K^{-1} = \begin{pmatrix} M_K^{-1} & \\ & R_\Omega M_K^{-1} R_\Omega \end{pmatrix}, \tag{21}$$

where $R_\Omega \in \mathbb{R}^{(n_{\text{inn}}/2)\times(n_{\text{inn}}/2)}$ is the reflection matrix for the inner unknowns and $M_K^{-1} \simeq K^{-1}$.

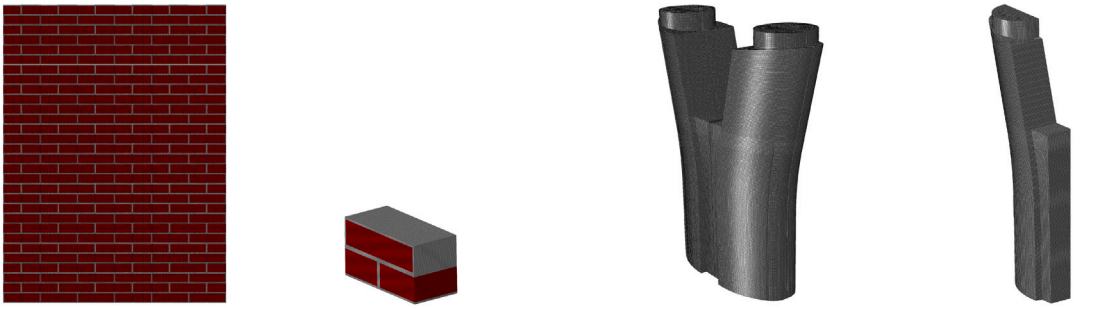
Then, similarly to Eq. (14), we aim to exploit the structure of $\bar{K}$ and $\bar{M}_K^{-1}$ to replace `SpMV` with the more efficient `SpMM` (combined with a specialised kernel for applying the reflections whose cost is negligible). As with the periodic structures, this will allow for a reduction of the memory footprint and setup costs since AMGR does not require $\bar{K}$ and $\bar{M}_K^{-1}$ but the smaller K and M_K^{-1}. Furthermore, the procedure above can be applied recursively to account for an arbitrary number of symmetries, s, therefore decomposing $\bar{K}$ into $n_b = 2^s$ blocks and strengthening the advantages of AMGR. Additionally, this can be done regardless of the boundary conditions, which can be moved to the interface block and, therefore, do not affect $\bar{K}$'s structure.

3.3. Tackling non-linearities

Non-linear problems are very common in Finite Element analyses and are typically solved through incremental Newton schemes. These methods require the repeated solution of linear systems whose stiffness matrix may vary from one iteration to the next, forcing the recomputation of the preconditioner at every non-linear step. As a result, all the AMG operators need to be reconstructed, and the setup costs become predominant. AMGR can also be applied to accelerate the solution of non-linear problems in which the non-linearity only affects a small part of the domain, something occurring very often in mechanics, *e.g.*, in contact problems, breaking tests or geomechanics.

Under such circumstances, AMGR greatly reduces the setup costs of AMG. It is enough to set as interface all the unknowns affected by the non-linearity, see Fig. 4. Then, $\bar{K}$ becomes constant, and the only parts of the system that we need to update are $\bar{B}$ and $\bar{C}$. Hence, the preconditioner setup costs are reduced approximately by a factor of $n_{\text{ifc}}/n_{\text{inn}}$.

Similar considerations apply to the reconstruction of the preconditioner, which only needs to update the smoother of the interface block, $\bar{M}_C^{-1} \simeq \bar{C}^{-1}$, the reduced operator, A_c, and its AMG approximation, $M_c^{-1} \simeq A^{-1}$.



(a) Masonry wall (b) Base masonry wall (c) Bridge pier (d) Base bridge pier

Fig. 5. (Left) Test problem for substructuring: masonry wall model. (Right) Test problem for reflection symmetries: bridge pier model.

4. Numerical results

This section investigates the advantages of AMGR in real-world applications. The problems considered cover all the strategies described in Section 3 and consist of the simulation of a masonry wall, a bridge pier, a coaxial propeller, a geophysical and a biomechanical model. While the former two allow for leveraging substructuring and reflection symmetries, the latter illustrate the gains in non-linear analysis.

For simplicity, the right-hand side (RHS) used in the tests is a random vector, which ensures a wide range of frequencies in the resulting residual, enabling a complete assessment of the preconditioner's effectiveness in damping all error components. All the executions rely on a combined MPI and OpenMP parallelism and have been conducted on the JFF cluster at the Heat and Mass Transfer Technological Center. Its non-uniform memory access (NUMA) nodes are equipped with two Intel Xeon 6230 CPUs (20 cores, 2.1 GHz, 27.5 MB L3 cache and 140 GB/s memory bandwidth) linked to 288 GB of RAM and interconnected through 7 GB/s FDR Infiniband.

4.1. AMGR and space regularities

Let us start by studying the performance of AMGR on problems presenting spatial regularities, which allow for increasing the arithmetic intensity of AMG by replacing SpMV with SpMM. Fig. 5 presents the Wall and Pier models, which arise from repetition and mirroring, respectively, of a *base model*.

As discussed in Sections 3.1 and 3.2, AMGR introduces an aggressive coarsening to the top level of the multigrid hierarchy. It relies on the population described in Algorithm 3.1, which guarantees a prescribed maximum interpolation distance, k. While $k = 1$ allows for an accurate interpolation, it also reduces the effectiveness of the reduction as the initial coarsening becomes overpopulated. In fact, Fig. 6 indicates that the resulting coarsening is less effective than that of the underlying AMG, as it leads to larger and denser coarse-grid operators. Conversely, despite making AMGR lighter, using larger values of k rapidly exceeds the range of applicability of long-distance interpolation formulas, such as Extended+I (ExtI) [79] or DPLS [80]. Indeed, the number of iterations grows fast with k, rendering the AMG reduction ineffective and evincing the need for an optimal k.

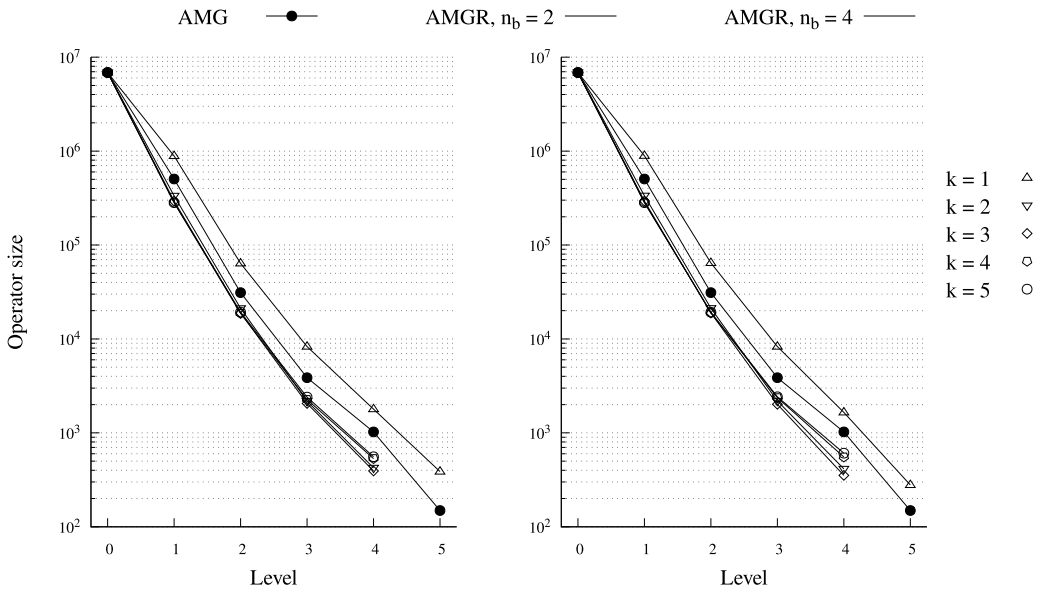
All the numerical experiments agreed with Fig. 6 in confirming $k = 2$ as the best-performing parameter. Firstly, it leads to a coarsening roughly as aggressive as the ones obtained with larger values of k (see Fig. 6(a)). Secondly, the resulting multigrid hierarchy is significantly lighter thanks to preventing the extra fill-in introduced by DPLS for larger interpolation distances (see Fig. 6(b)). Thirdly, as summarised in Table 1, the quality of the preconditioner does not deteriorate, converging $k = 2$ as fast as $k = 1$ (or the standard AMG). On top of these, the population algorithm remains cost-effective, as computing the symbolic power of T^2 in Algorithm 3.1 represents a minor overhead.

As described in Sections 3.1 and 3.2, in the presence of spatial regularities, AMGR not only reduces the setup costs and memory footprint of the top-level smoother but also accelerates its application. Both effects can be observed in the results of Table 2. On the one hand, $n_b = 2$ is enough to make AMGR's setup significantly faster than AMG's, becoming greater the gains as the number of subdomains increase. Regarding the application costs, we observe that, for $n_b = 2$, the solution (per iteration) of AMG is slightly faster than that of AMGR. This slight overhead can be understood by recalling our implementation of the matrix multiplication by the coefficient matrix, which, for the case of substructuring, *i.e.*, with $\bar{K} = \mathbb{I}_{n_b} \otimes K$, reads:

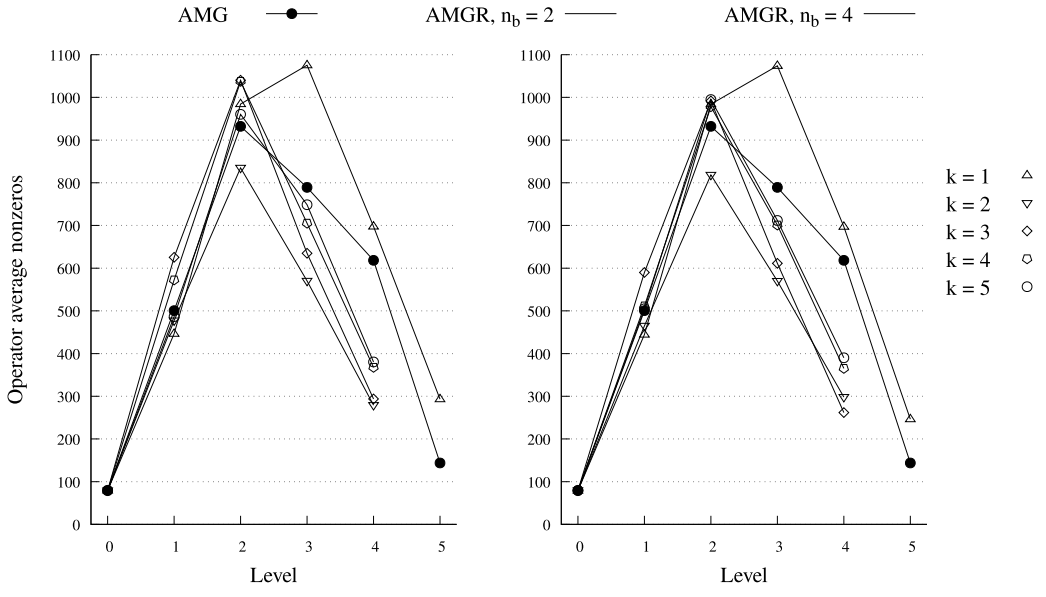
$$\begin{pmatrix} y_{\text{inn}} \\ y_{\text{ifc}} \end{pmatrix} = \begin{pmatrix} \bar{K} & \bar{B} \\ \bar{B}^T & \bar{C} \end{pmatrix} \begin{pmatrix} x_{\text{inn}} \\ x_{\text{ifc}} \end{pmatrix} = \begin{pmatrix} \texttt{SpMM}(K, x_{\text{inn}}) + \texttt{SpMV}(\bar{B}, x_{\text{ifc}}) \\ \texttt{SpMV}(\bar{B}^T, x_{\text{inn}}) + \texttt{SpMV}(\bar{C}, x_{\text{ifc}}) \end{pmatrix} \in \mathbb{R}^n, \tag{22}$$

where, for example, $\texttt{SpMM}(K, x_{\text{inn}}) \in \mathbb{R}^{n_{\text{inn}}}$ denotes the output of an SpMM between K and x_{inn}.

Considering that $n_{\text{inn}} \gg n_{\text{ifc}}$, it is clear that in Eq. (22), most computational efforts are devoted to the matrix multiplication by $\bar{K}$. While it is significantly accelerated thanks to replacing $\texttt{SpMV}(\bar{K}, x_{\text{inn}})$ with the more compute-intensive $\texttt{SpMM}(K, x_{\text{inn}})$, applying Eq. (22) introduces a slight overhead compared to the standard SpMV by the coefficient matrix, $\texttt{SpMV}(A, x) \in \mathbb{R}^n$. Indeed, the four



(a) Operator number of rows



(b) Operator average nonzeros per row

Fig. 6. Influence of the maximum interpolation distance on the multigrid hierarchy. Results obtained on the Pier problem using $n_b \in \{2,4\}$ and a varying maximum interpolation distance, $k \in \{1,\dots,5\}$.

separate kernel calls (together with the combination of their output) require two additional vector accesses. Despite SpMM gains for $n_b = 2$ not being enough to hide such extra overheads, increasing n_b, and therefore SpMM accelerations, make AMGR iterations faster than AMG's. According to Table 2, this is already the case for $n_b = 4$, which yielded up to 43% overall speed-ups.

It is worth noting that such speed-ups are possible thanks to AMGR not harming AMG's excellent rates of convergence. Fig. 7 illustrates this in the two model problems.

Let us also note that increasing the arithmetic intensity of AMG by replacing $\texttt{SpMV}(\bar{K}, x_{\text{inn}})$ with $\texttt{SpMM}(K, x_{\text{inn}})$ requires a consistent computational partitioning. Namely, to distribute the *base model* among the available resources and extend such a partitioning to the remaining subdomains. Fig. 8 exemplifies this by distributing an arbitrary 2D problem with two reflection symmetries (and, therefore, $n_b = 4$ subdomains) among four computing devices.

Of course, in the limit case of distributing small subdomains among lots of devices, such partitioning may be suboptimal. However, this is not an issue for realistically sized problems. Fig. 9 displays the strong scalability of AMG and AMGR on the Pier

Table 1
Overall influence of the maximum interpolation distance on AMGR.

preconditioner	n_b	k	coarsening ratio	avg nnzr	its
AMG	–	–	0.07	500.5	78
AMGR	2	1	0.13	446.5	81
AMGR		2	0.05	477.3	82
AMGR		3	0.04	625.4	84
AMGR		4	0.04	572.3	114
AMGR		5	0.04	486.5	1179
AMGR	4	1	0.13	444.7	80
AMGR		2	0.05	465.0	81
AMGR		3	0.04	589.8	84
AMGR		4	0.04	511.4	149
AMGR		5	0.04	505.0	1271

Table 2
AMGR results with substructuring (Wall) and reflection symmetries (Pier).

(a) Wall (n = 6.9e+6, $\kappa(A)$ = 1.4e+6, on two JFF nodes)

preconditioner	n_b	coarsening ratio	avg nnzr	its	t-pc (s)	t-sol (s)	t-tot (s)	speed-up
AMG	–	0.08	501.0	80	68.2	8.8	77.0	1.0x
AMGR	2	0.04	399.0	95	47.3	11.4	58.7	1.3x
AMGR	4	0.04	383.4	94	45.0	8.9	53.9	1.4x

(b) Pier (n = 6.9e+6, $\kappa(A)$ = 1.7e+6, on two JFF nodes)

preconditioner	n_b	coarsening ratio	avg nnzr	its	t-pc (s)	t-sol (s)	t-tot (s)	speed-up
AMG	–	0.07	501.0	78	78.3	10.4	88.7	1.0x
AMGR	2	0.05	477.3	82	54.4	11.7	66.1	1.3x
AMGR	4	0.05	465.0	81	53.9	9.2	63.1	1.4x

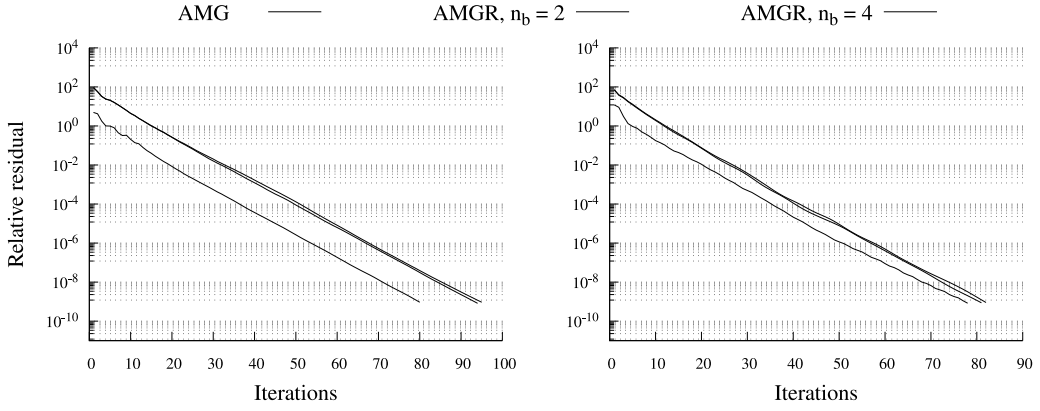


Fig. 7. Convergence of AMG and AMGR on the Wall (left) and Pier (right) models.

model. Despite the minimal decay for $n_b = 4$, AMG and AMGR exhibit almost identical scalability and run efficiently with workloads down to 40K dofs per CPU core, matching past strong scalability results of Chronos' AMG [75].

4.2. AMGR and non-linearities

When it comes to non-linear analysis, AMGR proves very effective in reducing AMG's setup costs, which typically account for 70% to 80% of the overall solution time. Fig. 10 presents the three models considered: a coaxial propeller, the geophysical model PUNQ [81] and a more ill-conditioned biomechanical model of a human heel [82]. They provide typical examples of localised non-linearities. The coaxial propeller exhibits a non-linearity due to contact, whereas a material non-linearity characterises the geophysical and biomechanical models.

As with spatial regularities, the best-performing interpolation distance for the population algorithm was $k = 2$. Table 3 compares the results obtained with AMG and AMGR. The AMG reduction was slightly more aggressive than AMG's coarsening on the coaxial propeller. However, this was not the case for the geophysical and biomechanical models, where AMGR's coarsening ratios were approximately 1.5 times larger. Despite this, the resulting reduced operators were about 5 times lighter, leading to about 30% faster iterations.

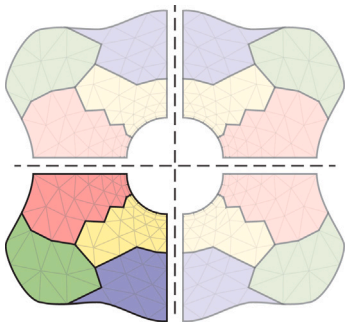


Fig. 8. Computational domain partitioning of a 2D model with two reflection symmetries.

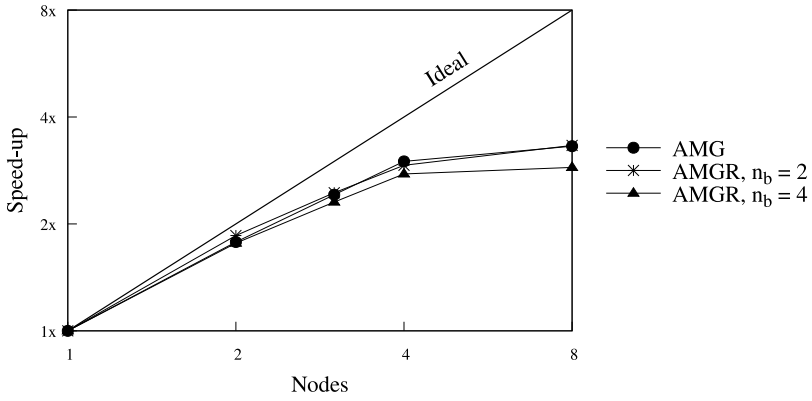
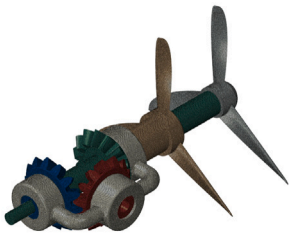
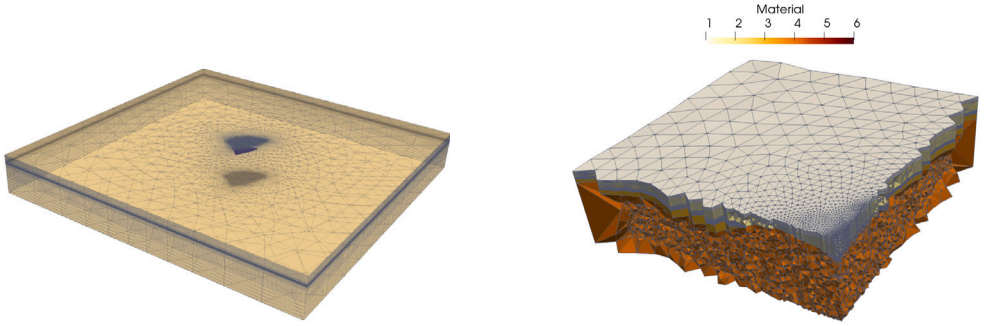


Fig. 9. Strong scalability analysis of AMG and AMGR on the Pier model problem. Workload ranging from 171K to 21K dofs per core.



(a) Coaxial propeller

(b) Biomechanical model



(c) Geophysical model

Fig. 10. Test problems for non-linear analysis.

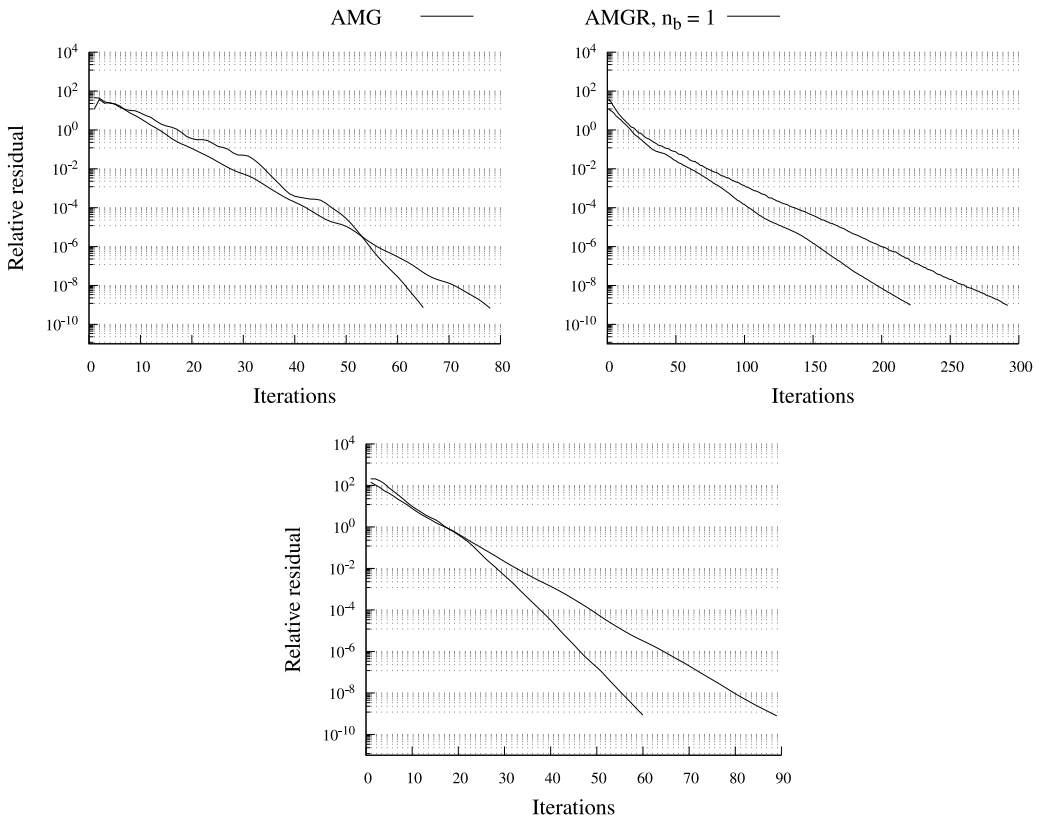


Fig. 11. Convergence of AMG and AMGR on the Coaxial propeller (top-left), geophysical (top-right) and biomechanical (bottom) models.

In the absence of spatial regularities, it is no longer possible to leverage $\bar{K}$'s structure to replace SpMV with SpMM. However, we still have to incur the overhead of applying the coefficient matrix through four separate kernel calls, as in Eq. (22), making AMGR iterations on the Coaxial propeller slightly slower. On top of that, as illustrated in Fig. 11, AMGR's convergence is comparable but a little slower than AMG's.

Despite such a priori discouraging results, AMGR was most effective in accelerating the solution of non-linear problems. As was already pointed out, for such cases, the preconditioner needs to be recomputed prior to each solution, and its reconstruction needs to be accounted for within the overall solution time. As discussed in Section 3.3, most of AMGR's setup can be recycled between non-linear iterations, only requiring to recompute the smoother of the interface block, $\bar{M}_C^{-1} \simeq \bar{C}^{-1}$, the reduced operator, A_c, and its AMG approximation, $M_c^{-1} \simeq A^{-1}$. Fig. 12 decomposes the portion of setup AMGR recycles, labelled as "Pre-processing", and the portion it needs recomputing at each non-linear iteration, labelled as "Setup". Two things become clear. First, in non-linear analysis, most of the overall solution time is devoted to the preconditioner's setup. And secondly, AMGR greatly reduces such time, yielding 3.7x overall speed-ups.

5. Conclusions

This research focuses on developing faster solution methods for structural problems. In particular, we presented an AMG reduction framework that enhances the standard AMG preconditioner by exploiting features commonly present in structural problems. While similar techniques have been used for scalar-valued Poisson's problems in the context of CFD simulations, this work reformulates them for vector-valued fields, like displacements in structural engineering, as well as for non-linear analysis. The features leveraged by AMGR, the resulting preconditioner, are periodicities, spatial symmetries, and localised non-linearities. In all such cases, AMGR introduces an AMG reduction to the top level of the multigrid hierarchy that makes it significantly lighter than a standard AMG. This is done by selecting a few unknowns to form an initial coarse space that is then expanded through an aggressive coarsening strategy. Subsequent levels are purely algebraic.

AMGR identifies the unknowns for the initial coarsening based on the problem's characteristics, granting its flexibility and allowing for further improvements. When applied to problems with periodicities or spatial symmetries, *i.e.*, decomposable into n_b subdomains, AMGR imposes a consistent ordering that makes the top-level smoother satisfy a constant block diagonal structure. This allows for replacing the standard SpMV with the more compute-intensive SpMM, therefore accelerating the smoother application and reducing its memory footprint and setup costs by a factor of n_b. Hence, the larger the number of subdomains, n_b, the greater

Table 3
AMGR results with non-linear analysis.

(a) Coaxial propeller (n = 4.6e+6, $\kappa(A)$ = 1.2e+8, on two JFF nodes)								
preconditioner	n_b	coarsening ratio	avg nnzr	its	t-pc (s)	t-sol (s)	t-tot (s)	speed-up
AMG	–	0.06	324.5	65	76.9	10.0	76.9	1.0x
AMGR	1	0.05	323.1	80	20.3	13.2	33.5	2.3x
(b) Geophysical model (n = 7.0e+5, $\kappa(A)$ = 2.6e+8, on one JFF node)								
preconditioner	n_b	coarsening ratio	avg nnzr	its	t-pc (s)	t-sol (s)	t-tot (s)	speed-up
AMG	–	0.07	439.6	221	13.3	5.5	18.8	1.0x
AMGR	1	0.12	77.3	292	2.5	5.6	8.1	2.3x
(c) Biomechanical model (n = 1.1e+6, $\kappa(A)$ = 1.6e+10, on one JFF node)								
preconditioner	n_b	coarsening ratio	avg nnzr	its	t-pc (s)	t-sol (s)	t-tot (s)	speed-up
AMG	–	0.06	348.4	61	17.7	2.1	19.8	1.0x
AMGR	1	0.08	80.7	89	3.1	2.3	5.4	3.7x

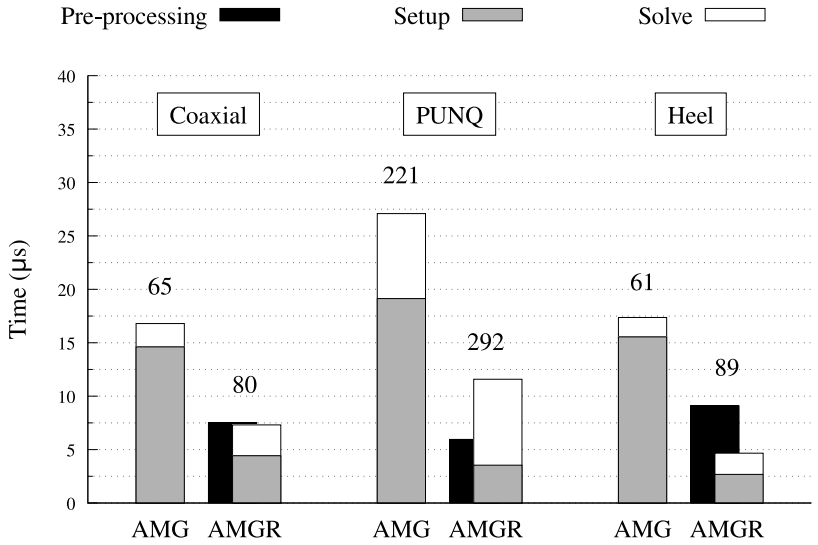


Fig. 12. Decomposition of the overall solution time normalised by the number of unknowns. While the setup phase is done at each non-linear iteration, the pre-processing stage is only performed once and, therefore, excluded from the total solution time. The number of iterations is indicated on top of each bar.

SpMM's advantages. Remarkably enough, this strategy can be applied regardless of the boundary conditions, as they do not affect the matrices' structure. Numerical experiments on a masonry wall and a bridge pier confirmed AMGR's capacity to leverage periodicities and spatial symmetries, respectively, being up to 43% faster than the standard AMG with $n_b = 4$ subdomains.

AMGR proved particularly effective on structural problems with non-linearities affecting only a smaller part of the domain. This is common in applications like contact mechanics or geomechanical simulations, for which the coefficient matrix changes at each non-linear iteration. Consequently, the preconditioner needs to be updated accordingly, and its reconstruction accounts for 70% to 80% of the overall solution time. AMGR removes the (few) unknowns affected by the non-linearity, making most of the preconditioner constant and greatly reducing the portion of the preconditioner that needs to be rebuilt. Numerical experiments on a coaxial propeller, a geophysical and a biomechanical simulation proved the advantages of AMGR for tackling localised non-linearities, yielding up to 3.7x overall speed-ups.

Hence, all results confirm that AMGR is very advantageous for solving structural problems with periodicities, reflection symmetries, or localised non-linearities. However, to use it, the simulation codes must be adapted appropriately, as AMGR cannot infer from the coefficient matrix which features to leverage. Localised non-linearities are relatively simple to manage by placing last the dofs affected by the non-linearity (and therefore changing from a non-linear iteration to the next). On the other hand, leveraging periodicities and symmetries may be more challenging due to the discrete model having to preserve them. While this can be easily accomplished on in-house codes, *e.g.*, by only discretising one of the subdomains and managing the interfaces separately, it may be more challenging on third-party simulation codes.

Future developments include the implementation of AMGR for graphics processing units (GPUs), which will greatly benefit from the higher arithmetic intensity of SpMM. Additionally, we aim to extend the AMG reduction framework on which AMGR is based towards general problems. Despite not leveraging the problem's characteristics, the resulting strategy would benefit from the aggressive coarsening it introduces.

CRediT authorship contribution statement

Àdel Alsalti-Baldellou: Writing – original draft, Software, Investigation, Conceptualization. **Andrea Franceschini:** Writing – original draft, Investigation, Conceptualization. **Gianluca Mazzucco:** Visualization, Investigation. **Carlo Janna:** Writing – original draft, Supervision, Software, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was partly funded by the project "National Centre for HPC, Big Data and Quantum Computing", CN00000013 (approvato nell'ambito del Bando M42C – Investimento 1.4 – Avvisto "Centri Nazionali" – D.D. n. 3138 del 16.12.2021, ammesso a finanziamento con Decreto del MUR n. 1031 del 17.06.2022). Portions of this work were also supported by the 2024 INdAM-GNCS project "Advanced numerical methods for poromechanics: theoretical properties and computational aspects". Numerical experiments were conducted on the JFF cluster at the Heat and Mass Transfer Technological Center. The authors thankfully acknowledge these institutions.

References

[1] J. Maljaars, D. Leonetti, B. Hashemi, H.B. Snijder, Systematic derivation of safety factors for the fatigue design of steel bridges, Struct. Saf. 97 (2022) 102229, http://dx.doi.org/10.1016/j.strusafe.2022.102229.
[2] H. Xiao, L. Luo, J. Shi, H. Jiang, Z. Wu, Stressing state analysis of multi-span continuous steel-concrete composite box girder, Eng. Struct. 246 (2021) 113070, http://dx.doi.org/10.1016/j.engstruct.2021.113070.
[3] I. Shakir, M.A. Jasim, S.S. Weli, High rise buildings: Design, analysis, and safety: An overview, Int. J. Archit. Eng. Technol. 8 (2021) 1–13, http://dx.doi.org/10.15377/2409-9821.2021.08.1.
[4] S. Ereiz, I. Duvnjak, J.F. Jiménez-Alonso, Review of finite element model updating methods for structural applications, in: Structures, Vol. 41, Elsevier, 2022, pp. 684–723, http://dx.doi.org/10.1016/j.istruc.2022.05.041.
[5] I. Koutromanos, A. Stavridis, P.B. Shing, K. Willam, Numerical modeling of masonry-infilled RC frames subjected to seismic loads, Comput. Struct. 89 (11–12) (2011) 1026–1037, http://dx.doi.org/10.1016/j.compstruc.2011.01.006.
[6] R. Mudragada, P. Bhargava, Masonry-infilled RC frames exposed to blast loads: A review on numerical modeling and response, Pract. Period. Struct. Des. Constr. 27 (4) (2022) 03122006, http://dx.doi.org/10.1061/(ASCE)SC.1943-5576.0000727.
[7] F.-F. Feng, H.-J. Hwang, Y. Zhou, J.-M. Sun, H.-Z. Zhang, J.-H. Roh, S.-M. Kang, W.-J. Yi, Effect of three-dimensional space on progressive collapse resistance of reinforced concrete frames under various column removal scenarios, J. Build. Eng. 90 (2024) 109405, http://dx.doi.org/10.1016/j.jobe.2024.109405.
[8] M. Yazdani, Three-dimensional nonlinear finite element analysis for load-carrying capacity prediction of a railway arch bridge, Int. J. Civ. Eng. 19 (7) (2021) 823–836, http://dx.doi.org/10.1007/s40999-021-00608-w.
[9] G. Wang, Y. Zhang, Z. Zheng, H. Chen, J. Yu, Crashworthiness design and impact tests of aluminum foam-filled crash boxes, Thin-Walled Struct. 180 (2022) 109937, http://dx.doi.org/10.1016/j.tws.2022.109937.
[10] Y. Ma, Z. Guo, L. Wang, J. Zhang, Probabilistic life prediction for reinforced concrete structures subjected to seasonal corrosion-fatigue damage, J. Struct. Eng. 146 (7) (2020) 04020117, http://dx.doi.org/10.1061/(ASCE)ST.1943-541X.0002666.
[11] A.P. Vassilopoulos, The history of fiber-reinforced polymer composite laminate fatigue, Int. J. Fatigue 134 (2020) 105512, http://dx.doi.org/10.1016/j.ijfatigue.2020.105512.
[12] M.O. Heller, Finite element analysis in orthopedic biomechanics, in: Human Orthopaedic Biomechanics, Elsevier, 2022, pp. 637–658, http://dx.doi.org/10.1016/B978-0-12-824481-4.00026-3.
[13] N. Petitjean, P. Canadas, P. Royer, D. Noël, S. Le Floc'h, Cartilage biomechanics: From the basic facts to the challenges of tissue engineering, J. Biomed. Mater. Res. A 111 (7) (2023) 1067–1089, http://dx.doi.org/10.1002/jbm.a.37478.
[14] M.S. Khan, S. Shahabad, M. Yavuz, W. Duley, E. Biro, Y. Zhou, Numerical modelling and experimental validation of the effect of laser beam defocusing on process optimization during fiber laser welding of automotive press-hardened steels, J. Manuf. Process. 67 (2021) 535–544, http://dx.doi.org/10.1016/j.jmapro.2021.05.006.
[15] D. Connolly, G. Sivaswamy, S. Rahimi, V. Vorontsov, Miniaturised experimental simulation of open-die forging, J. Mater. Res. Technol. 26 (2023) 3146–3161, http://dx.doi.org/10.1016/j.jmrt.2023.08.073.
[16] S. Wang, T.J. Larsen, H. Bredmose, Ultimate load analysis of a 10 MW offshore monopile wind turbine incorporating fully nonlinear irregular wave kinematics, Mar. Struct. 76 (2021) 102922, http://dx.doi.org/10.1016/j.marstruc.2020.102922.
[17] O. Axelsson, Iterative Solution Methods, Cambridge University Press, 1996, http://dx.doi.org/10.1017/CBO9780511624100.
[18] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, 2003, http://dx.doi.org/10.1137/1.9780898718003.
[19] H.A. Van der Vorst, Iterative Krylov Methods for Large Linear Systems, in: Cambridge Monographs on Applied and Computational Mathematics, (13) Cambridge University Press, 2003, http://dx.doi.org/10.1017/CBO9780511615115.
[20] T.B. Jönsthövel, M.B. van Gijzen, C. Vuik, A. Scarpas, On the use of rigid body modes in the deflated preconditioned conjugate gradient method, SIAM J. Sci. Comput. 35 (1) (2013) B207–B225, http://dx.doi.org/10.1137/100803651.
[21] A. Gerstenberger, R.S. Tuminaro, An algebraic multigrid approach to solve extended finite element method based fracture problems, Internat. J. Numer. Methods Engrg. 94 (3) (2013) 248–272, http://dx.doi.org/10.1002/nme.4442.
[22] S. Gratton, S. Mercier, N. Tardieu, X. Vasseur, Limited memory preconditioners for symmetric indefinite problems with application to structural mechanics, Numer. Linear Algebra Appl. 23 (5) (2016) 865–887, http://dx.doi.org/10.1002/nla.2058.

[23] Y. Saad, H.A. Van Der Vorst, Iterative solution of linear systems in the 20th century, J. Comput. Appl. Math. 123 (1–2) (2000) 1–33, http://dx.doi.org/10.1016/S0377-0427(00)00412-X.
[24] M. Benzi, Preconditioning techniques for large linear systems: a survey, J. Comput. Phys. 182 (2) (2002) 418–477, http://dx.doi.org/10.1006/jcph.2002.7176.
[25] J.A. Meijerink, H.A. Van Der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, Math. Comput. 31 (137) (1977) 148–162, http://dx.doi.org/10.1090/S0025-5718-1977-0438681-4.
[26] T.F. Chan, H.A. Van Der Vorst, Approximate and incomplete factorizations, in: Parallel Numerical Algorithms, Springer, 1997, pp. 167–202, http://dx.doi.org/10.1007/978-94-011-5412-3_6.
[27] Y. Saad, ILUM: A multi-elimination ILU preconditioner for general sparse matrices, SIAM J. Sci. Comput. 17 (4) (1996) 830–847, http://dx.doi.org/10.1137/0917054.
[28] J. Berns-Müller, A. Spence, Inexact inverse iteration with variable shift for nonsymmetric generalized eigenvalue problems, SIAM J. Matrix Anal. Appl. 28 (4) (2006) 1069–1082, http://dx.doi.org/10.1137/050623255.
[29] M. Benzi, C.D. Meyer, M. Tuma, A sparse approximate inverse preconditioner for the conjugate gradient method, SIAM J. Sci. Comput. 17 (5) (1996) 1135–1149, http://dx.doi.org/10.1137/S1064827594271421.
[30] M. Benzi, M. Tuma, A sparse approximate inverse preconditioner for nonsymmetric linear systems, SIAM J. Sci. Comput. 19 (3) (1998) 968–994, http://dx.doi.org/10.1137/S1064827595294691.
[31] M. Benzi, M. Tuma, A comparative study of sparse approximate inverse preconditioners, Appl. Numer. Math. 30 (2–3) (1999) 305–340, http://dx.doi.org/10.1016/S0168-9274(98)00118-4.
[32] M.J. Grote, T. Huckle, Parallel preconditioning with sparse approximate inverses, SIAM J. Sci. Comput. 18 (3) (1997) 838–853, http://dx.doi.org/10.1137/S1064827594276552.
[33] L.Y. Kolotilina, A.Y. Yeremin, Factorized sparse approximate inverse preconditionings I. Theory, SIAM J. Matrix Anal. Appl. 14 (1) (1993) 45–58, http://dx.doi.org/10.1137/0614004.
[34] C. Janna, M. Ferronato, F. Sartoretto, G. Gambolati, FSAIPACK: A software package for high-performance factored sparse approximate inverse preconditioning, ACM Trans. Math. Softw. 41 (2) (2015) 1–26, http://dx.doi.org/10.1145/2629475.
[35] S. Koric, Q. Lu, E. Guleryuz, Evaluation of massively parallel linear sparse solvers on unstructured finite element meshes, Comput. Struct. 141 (C) (2014) 19–25, http://dx.doi.org/10.1016/j.compstruc.2014.05.009.
[36] S. Koric, A. Gupta, Sparse matrix factorization in the implicit finite element method on petascale architecture, Comput. Methods Appl. Mech. Engrg. 302 (2016) 281–292, http://dx.doi.org/10.1016/j.cma.2016.01.011.
[37] M. Adams, Evaluation of three unstructured multigrid methods on 3D finite element problems in solid mechanics, Internat. J. Numer. Methods Engrg. 55 (5) (2002) 519–534, http://dx.doi.org/10.1002/nme.506.
[38] F.J. Gaspar, C. Rodrigo, On the fixed-stress split scheme as smoother in multigrid methods for coupling flow and geomechanics, Comput. Methods Appl. Mech. Engrg. 326 (2017) 526–540, http://dx.doi.org/10.1016/j.cma.2017.08.025.
[39] J.W. Ruge, K. Stüben, Algebraic multigrid, in: Multigrid Methods, SIAM, 1987, pp. 73–130, http://dx.doi.org/10.1137/1.9781611971057.ch4.
[40] K. Stüben, A review of algebraic multigrid, in: Numerical Analysis: Historical Developments in the 20th Century, Elsevier, 2001, pp. 331–359, http://dx.doi.org/10.1016/B978-0-444-50617-7.50015-X.
[41] M. Adams, R. Taylor, Parallel multigrid solvers for 3D-unstructured large deformation elasticity and plasticity finite element problems, Finite Elem. Anal. Des. 36 (3) (2000) 197–214, http://dx.doi.org/10.1016/S0168-874X(00)00033-0, Robert J. Melosh Medal Competition, Duke University, Durham NC, USA, March 1999.
[42] R.D. Falgout, U.M. Yang, hypre: A library of high performance preconditioners, in: International Conference on Computational Science, Springer, 2002, pp. 632–641, http://dx.doi.org/10.1007/3-540-47789-6_66.
[43] The Trilinos Project Team, The trilinos project website, 2020, URL https://trilinos.github.io, (acccessed May 22, 2020).
[44] S. Balay, S. Abhyankar, M.F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, J. Faibussowitsch, W.D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M.G. Knepley, F. Kong, S. Kruger, D.A. May, L.C. McInnes, R.T. Mills, L. Mitchell, T. Munson, J.E. Roman, K. Rupp, P. Sanan, J. Sarich, B.F. Smith, S. Zampini, H. Zhang, H. Zhang, J. Zhang, PETSc/TAO users manual, Tech. Rep. ANL-21/39 - Revision 3.21, Argonne National Laboratory, 2024, http://dx.doi.org/10.2172/2205494.
[45] A. Abdelfattah, H. Anzt, E.G. Boman, E. Carson, T. Cojean, J. Dongarra, A. Fox, M. Gates, N.J. Higham, X.S. Li, J. Loe, P. Luszczek, S. Pranesh, S. Rajamanickam, T. Ribizel, B.F. Smith, K. Swirydowicz, S. Thomas, S. Tomov, Y.M. Tsai, U.M. Yang, A survey of numerical linear algebra methods utilizing mixed-precision arithmetic, Int. J. High Perform. Comput. Appl. 35 (2021) 344–369, http://dx.doi.org/10.1177/10943420211003313.
[46] H. Ibeid, L. Olson, W.D. Gropp, FFT, FMM, and multigrid on the road to exascale: performance challenges and opportunities, J. Parallel Distrib. Comput. 136 (2020) 63–74, http://dx.doi.org/10.1016/j.jpdc.2019.09.014.
[47] C. Fu, T. Yang, Sparse LU factorization with partial pivoting on distributed memory machines, in: Proceedings of the 1996 ACM/IEEE Conference on Supercomputing, 1996, pp. 31–es, http://dx.doi.org/10.1145/369028.369092.
[48] P. Nayak, T. Cojean, H. Anzt, Evaluating asynchronous Schwarz solvers on GPUs, Int. J. High Perform. Comput. Appl. 35 (3) (2021) 226–236, http://dx.doi.org/10.1177/1094342020946814.
[49] P. Sanan, S.M. Schnepp, D.A. May, Pipelined, flexible Krylov subspace methods, SIAM J. Sci. Comput. 38 (5) (2016) C441–C470, http://dx.doi.org/10.1137/15M1049130.
[50] P. Amestoy, C. Ashcraft, O. Boiteau, A. Buttari, J.-Y. L'Excellent, C. Weisbecker, Improving multifrontal methods by means of block low-rank representations, SIAM J. Sci. Comput. 37 (2015) A1451–A1474, http://dx.doi.org/10.1137/120903476.
[51] P. Ghysels, X.S. Li, F.H. Rouet, S. Williams, A. Napov, An efficient multicore implementation of a novel HSS-structured multifrontal solver using randomized sampling, SIAM J. Sci. Comput. 38 (2016) S358–S384, http://dx.doi.org/10.1137/15M1010117.
[52] R. Li, Y. Saad, Low-rank correction methods for algebraic domain decomposition preconditioners, SIAM J. Matrix Anal. Appl. 38 (2017) 807–828, http://dx.doi.org/10.1137/16M110486X.
[53] C.P. Kohar, L. Greve, T.K. Eller, D.S. Connolly, K. Inal, A machine learning framework for accelerating the design process using CAE simulations: An application to finite element analysis in structural crashworthiness, Comput. Methods Appl. Mech. Engrg. 385 (2021) 114008, http://dx.doi.org/10.1016/j.cma.2021.114008.
[54] J. He, S. Koric, S. Kushwaha, J. Park, D. Abueidda, I. Jasiuk, Novel DeepONet architecture to predict stresses in elastoplastic structures with variable complex geometries and loads, Comput. Methods Appl. Mech. Engrg. 415 (2023) 116277, http://dx.doi.org/10.1016/j.cma.2023.116277.
[55] A. Alsalti-Baldellou, X. Álvarez-Farré, A. Gorobets, F.X. Trias, Efficient strategies for solving the variable Poisson equation with large contrasts in the coefficients, in: 8th European Congress on Computational Methods in Applied Sciences and Engineering, Vol. 273, CIMNE, 2022, pp. 416–434, http://dx.doi.org/10.23967/eccomas.2022.107.
[56] A. Gorobets, F.X. Trias, R. Borrell, O. Lehmkuhl, A. Oliva, Hybrid MPI+OpenMP parallelization of an FFT-based 3D Poisson solver with one periodic direction, Comput. & Fluids 49 (1) (2011) 101–109, http://dx.doi.org/10.1016/j.compfluid.2011.05.003.
[57] O. Shishkina, A. Shishkin, C. Wagner, Simulation of turbulent thermal convection in complicated domains, J. Comput. Appl. Math. 226 (2009) 336–344, http://dx.doi.org/10.1016/j.cam.2008.08.008.

[58] Q.M. Bui, F.P. Hamon, N. Castelletto, D. Osei-Kuffuor, R.R. Settgast, J.A. White, Multigrid reduction preconditioning framework for coupled processes in porous and fractured media, Comput. Methods Appl. Mech. Engrg. 387 (2021) 114111, http://dx.doi.org/10.1016/j.cma.2021.114111.
[59] R.D. Falgout, S. Friedhoff, T.V. Kolev, S.P. MacLachlan, J.B. Schroder, Parallel time integration with multigrid, SIAM J. Sci. Comput. 36 (2014) C635–C661, http://dx.doi.org/10.1137/130944230.
[60] A. Alsalti-Baldellou, C. Janna, X. Álvarez-Farré, F.X. Trias, A multigrid reduction framework for domains with symmetries, 2024, arXiv:2407.05930.
[61] A. Alsalti-Baldellou, X. Álvarez-Farré, G. Colomer, A. Gorobets, C.D. Pérez-Segarra, A. Oliva, F.X. Trias, Lighter and faster simulations on domains with symmetries, Comput. & Fluids 275 (2024) 106247, http://dx.doi.org/10.1016/j.compfluid.2024.106247.
[62] H. Anzt, S. Tomov, J. Dongarra, On the performance and energy efficiency of sparse linear algebra on GPUs, Int. J. High Perform. Comput. Appl. 31 (2017) 375–390, http://dx.doi.org/10.1177/1094342016672081.
[63] A.J. Aref, K.M. Dolatshahi, A three-dimensional cyclic meso-scale numerical procedure for simulation of unreinforced masonry structures, Comput. Struct. 120 (2013) 9–23, http://dx.doi.org/10.1016/j.compstruc.2013.01.012.
[64] E. Tubaldi, L. Macorini, B.A. Izzuddin, Three-dimensional mesoscale modelling of multi-span masonry arch bridges subjected to scour, Eng. Struct. 165 (2018) 486–500, http://dx.doi.org/10.1016/j.engstruct.2018.03.031.
[65] K. Bi, H. Hao, Numerical simulation of pounding damage to bridge structures under spatially varying ground motions, Eng. Struct. 46 (2013) 62–76, http://dx.doi.org/10.1016/j.engstruct.2012.07.012.
[66] M. Chilakala, S. Garg, Three dimensional numerical simulation of cracks in an automobile wheel rim, Mater. Today: Proc. 26 (2020) 2032–2039, http://dx.doi.org/10.1016/j.matpr.2020.02.441.
[67] Y. Liu, Y. Zhao, M. Liu, X. Sun, A general parameterized high precision finite element modelling method of three-dimensional helical gear, in: IOP Conference Series: Materials Science and Engineering, Vol. 569, (3) IOP Publishing, 2019, 032041, http://dx.doi.org/10.1088/1757-899X/569/3/032041.
[68] S.-H. Ju, C.-Y. Fan, G. Wu, Three-dimensional finite elements of steel bolted connections, Eng. Struct. 26 (3) (2004) 403–413, http://dx.doi.org/10.1016/j.engstruct.2003.11.001.
[69] S. Li, Gear contact model and loaded tooth contact analysis of a three-dimensional, thin-rimmed gear, J. Mech. Des. 124 (3) (2002) 511–517, http://dx.doi.org/10.1115/1.1485290.
[70] M. Bagheri, A. Settari, Modeling of geomechanics in naturally fractured reservoirs, SPE Reserv. Eval. Eng. 11 (01) (2008) 108–118, http://dx.doi.org/10.2118/93083-PA.
[71] M. Ferronato, G. Gambolati, C. Janna, P. Teatini, Numerical modelling of regional faults in land subsidence prediction above gas/oil reservoirs, Int. J. Numer. Anal. Methods Geomech. 32 (6) (2008) 633–657, http://dx.doi.org/10.1002/nag.640.
[72] M.R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Natl. Bur. Stand. 49 (6) (1952) 409–436.
[73] Y. Saad, M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 856–869, http://dx.doi.org/10.1137/0907058.
[74] H.A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (2) (1992) 631–644, http://dx.doi.org/10.1137/0913035.
[75] G. Isotton, M. Frigo, N. Spiezia, C. Janna, Chronos: A general purpose classical AMG solver for High Performance Computing, SIAM J. Sci. Comput. 43 (5) (2021) C335–C357, http://dx.doi.org/10.1137/21M1398586.
[76] Chronos web page, 2020, URL https://www.m3eweb.it/chronos.
[77] A. Franceschini, V.A.P. Magri, G. Mazzucco, N. Spiezia, C. Janna, A robust adaptive algebraic multigrid linear solver for structural mechanics, Comput. Methods Appl. Mech. Engrg. 352 (2019) 389–416, http://dx.doi.org/10.1016/j.cma.2019.04.034.
[78] C. Janna, M. Ferronato, Adaptive pattern research for block FSAI preconditioning, SIAM J. Sci. Comput. 33 (2011) 3357–3380, http://dx.doi.org/10.1137/100810368.
[79] H.D. Sterck, R.D. Falgout, J.W. Nolting, U.M. Yang, Distance-two interpolation for parallel algebraic multigrid, Numer. Linear Algebra Appl. 15 (2008) 115–139, http://dx.doi.org/10.1002/nla.559.
[80] V.A. Paludetto Magri, A. Franceschini, C. Janna, A novel algebraic multigrid approach based on adaptive smoothing and prolongation for ill-conditioned systems, SIAM J. Sci. Comput. 41 (1) (2019) A190–A219, http://dx.doi.org/10.1137/17M1161178.
[81] G. Gao, M. Zafari, A.C. Reynolds, Quantifying uncertainty for the PUNQ-S3 problem in a Bayesian setting with RML and EnKF, SPE J. 11 (04) (2006) 506–515, http://dx.doi.org/10.2118/93324-PA.
[82] C. Fontanella, S. Matteoli, E. Carniel, J. Wilhjelm, A. Virga, A. Corvi, A. Natali, Investigation on the load-displacement curves of a human healthy heel pad: In vivo compression data compared to numerical results, Med. Eng. Phys. 34 (9) (2012) 1253–1259, http://dx.doi.org/10.1016/j.medengphy.2011.12.013.