ESAIM: M2AN
Vol. 40, N° 1, 2006, pp. 99–122
DOI: 10.1051/m2an:2006004

ESAIM: Mathematical Modelling and Numerical Analysis
www.edpsciences.org/m2an

A NUMERICAL STUDY ON NEUMANN-NEUMANN METHODS FOR *hp* APPROXIMATIONS ON GEOMETRICALLY REFINED BOUNDARY LAYER MESHES II. THREE-DIMENSIONAL PROBLEMS *

Andrea Toselli[1] and Xavier Vasseur[1]

Abstract. In this paper, we present extensive numerical tests showing the performance and robustness of a Balancing Neumann-Neumann method for the solution of algebraic linear systems arising from *hp* finite element approximations of scalar elliptic problems on geometrically refined boundary layer meshes in three dimensions. The numerical results are in good agreement with the theoretical bound for the condition number of the preconditioned operator derived in [Toselli and Vasseur, *IMA J. Numer. Anal.* **24** (2004) 123–156]. They confirm that the condition numbers are independent of the aspect ratio of the mesh and of potentially large jumps of the coefficients. Good results are also obtained for certain singularly perturbed problems. The condition numbers only grow polylogarithmically with the polynomial degree, as in the case of *p* approximations on shape-regular meshes [Pavarino, *RAIRO: Modél. Math. Anal. Numér.* **31** (1997) 471–493]. This paper follows [Toselli and Vasseur, *Comput. Methods Appl. Mech. Engrg.* **192** (2003) 4551–4579] on two dimensional problems.

Mathematics Subject Classification. 65N22, 65N35, 65N55.

Received: December 23, 2004. Revised: July 3 and September 8, 2005.

Introduction

In recent years *hp* finite element methods have gained an increasing popularity in both the applied mathematics community and some engineering application fields. The *hp* finite element method has been first introduced by Gui and Babuška [26] and since then some monographs [30, 37, 50, 54] or part of textbooks ([38], Sect. 8.4), have proposed both theoretical and numerical insights into this topic. These methods are found to be particularly useful when high or extremely high accuracy is needed and when minimal dissipation and dispersion errors in the discrete system are required.

Indeed the main reason for the interest in *hp* finite element methods is that they achieve exponential rates of convergence for *both* regular and singular solutions [37, 50]. In presence of singularities or boundary layers, suitably graded meshes, geometrically refined towards corners, edges and/or faces have to be employed to achieve such an exponential rate of convergence [37, 50]. Thus highly stretched meshes with huge aspect ratios are obtained in practice. Consequently, the condition number of the stiffness matrix severely deteriorates:

Keywords and phrases. Domain decomposition, preconditioning, *hp* finite elements, spectral elements, anisotropic meshes.

* *This work was partially supported by the Swiss National Science Foundation under Project 20-63397.00.*

[1] Seminar for Applied Mathematics, ETH Zürich, Rämistrasse 101, 8092 Zürich, Switzerland. Xavier.Vasseur@cerfacs.fr

an exponential growth in the spectral polynomial degree is obtained for the condition number of the stiffness matrix. Hence robust iterative solvers are mandatory especially for three-dimensional applications.

In this work, solvers based on domain decomposition methods of iterative substructuring type [43,53,59] will be considered.

Several works on domain decomposition have been proposed for higher order approximations [2,27,31,33,40] and more recently [32] (see also the references therein). Unfortunately, *up to now*, no iterative substructuring method has been proven to be efficient when very thin elements and/or subdomains (involving meshes with high aspect ratio) or general non quasiuniform meshes are employed. Thus our goal is to fill this gap by proposing a domain decomposition preconditioner that is robust with the mesh aspect ratio and possible large jumps in the coefficients. Its theoretical derivation has been presented in [58]. In this paper the main emphasis is devoted to an extensive numerical study of its performances. These robustness aspects will be carefully summarized and analyzed hereafter for some three-dimensional elliptic model problems of diffusion or reaction-diffusion type. These model problems defined on simple geometries have been chosen here in order to be easily reproducible. Nevertheless some of them do lead to very ill conditioned linear systems and are thus relevant for testing the numerical performances of a given preconditioner.

In [56, 57], we showed that the Balancing Neumann-Neumann [35] and one-level Finite Element Tearing and Interconnecting (FETI) [19] methods can be successfully devised for some particular anisotropic meshes commonly used for *hp* finite element approximations of two-dimensional problems. Indeed, these meshes are highly anisotropic, but of a particular type:

(1) they are obtained by refining an initial *shape-regular* mesh (macromesh);
(2) refinement is only carried *towards* the boundary of the computational domain.

These properties allowed us to obtain condition number bounds for the corresponding preconditioned operators that only grow polylogarithmically with the polynomial degree, as is the case of p approximations on shape regular meshes [41]. Our understanding and analysis was confirmed by numerical experiments [57]. Following similar ideas, we have been able to extend these results to three-dimensional problems in [58]. The main theoretical result of that work is that certain Balancing Neumann-Neumann methods provide condition numbers independent of the aspect ratio of the mesh and of potentially large jumps of the coefficients, still retaining a polylogarithmic growth in the number of unknowns. This analysis has been developed on a simple diffusion problem and few numerical experiments on a purely diffusive problem defined on a geometrically refined mesh have also been provided confirming the polylogarithmic growth of the condition number of the preconditioned operator in the polynomial degree. In this paper as already pointed we will treat more general diffusion problems with possibly large jumps in the coefficients or reaction-diffusion problems of singularly perturbed type and present extensive results for six different cases.

The remainder of this paper is organized as follows: in Section 1, we introduce the model problem for our proposed numerical study, the *hp* finite element approximations and finally a class of geometrically refined meshes. The proposed Balancing Neumann-Neumann domain decomposition preconditioner is described in Section 2. An extensive numerical study is presented in Section 3. We end this work by mentioning some perspectives and future developments in Section 4.

1. Problem setting and *hp* finite element approximations

1.1. Model problem

We consider a linear, elliptic problem on a bounded polyhedral domain $\Omega \subset \mathbb{R}^3$ of unit diameter, formulated variationally as:
Find $u \in H_0^1(\Omega)$, such that

$$a(u,v) = \int_\Omega (\varepsilon\, \rho(\mathbf{x}) \nabla u \cdot \nabla v + c\, u\, v)\, \mathrm{d}\mathbf{x} = f(v), \quad v \in H_0^1(\Omega), \tag{1}$$

where c, ε are non-negative real coefficients. As usual, $H^1(\Omega)$ is the space of square summable functions with square summable first derivatives, and $H^1_0(\Omega)$ its subspace of functions that vanish on $\partial\Omega$. The functional $f(\cdot)$ belongs to the dual space $H^{-1}(\Omega)$. Here $\mathbf{x} = (x, y, z)$ denotes the position vector.

The coefficient $\rho(\mathbf{x}) > 0$ can be discontinuous, with very different values for different subregions of Ω, but we allow it to vary only moderately within each subregion. Without decreasing the generality of our results, we will only consider the piecewise constant case *i.e.* $\rho(\mathbf{x}) = \rho_i, \quad \mathbf{x} \in \Omega_i$.

Note that the purely diffusion problem derived from (1) has been used in [58] as a model problem to derive the condition number bound for the Balancing Neumann-Neumann preconditioner. This bound will be recalled without proof in Section 2.4.

1.2. *hp* finite element approximations

We now specify a particular choice of finite element spaces. Let $\mathcal{T}$ be a mesh consisting of affinely mapped cubes. Given a polynomial degree $k \geq 1$, we consider the following finite element spaces:

$$X = X^k(\Omega; \mathcal{T}) = \left\{ u \in H^1_0(\Omega) \,|\quad u_{|_K} \in \mathbb{Q}_k(K),\ K \in \mathcal{T} \right\}. \tag{2}$$

Here $\mathbb{Q}_k(K)$ is the space of polynomials of maximum degree k in each variable on K. In the following, we may drop the reference to k, Ω, and/or $\mathcal{T}$ whenever there is no confusion.

In this work, interpolating Lagrange polynomials on Gauss-Lobatto nodes are used as a particular nodal basis of $X^k(\Omega; \mathcal{T})$. The set of Gauss-Lobatto points $GLL(k)$ is the set of (distinct and real) zeros of $(1 - x^2)L'_k(x)$, with L_k the Legendre polynomial of degree k (*cf.* [9], Sect. 3) and the quadrature formula based on $GLL(k)$ has order $2k-1$. In this work, quadrature formulas based on $GLL(k)$ are chosen. Given the nodes $GLL(k)^3$ on the reference element $\hat{Q} = (-1, 1)^3$, our basis functions on $\mathbb{Q}_k(\hat{Q})$ are defined as tensor products of kth order Lagrange interpolating polynomials on $GLL(k)$. More details on spectral element methods can be found in, *e.g.*, [9].

In this paper, we always assume that the meshes are *regular*, *i.e.*, the intersection between neighboring elements is either a vertex, or an edge, or a face that is common to both elements.

A finite element approximation of (1) consists of finding $u \in X$, such that

$$a(u, v) = f(v), \quad v \in X. \tag{3}$$

1.3. Geometric boundary layer meshes

We now introduce a class of geometrically graded meshes. They are determined by a mesh grading factor $\sigma \in (0, 1)$ and a refinement level $n \geq 0$. The number of layers is $n + 1$ and the thinnest layer has a width proportional to σ^n. Robust exponential convergence of *hp* finite element approximations is achieved if n is suitably chosen. For singularity resolution, n is required to be proportional to the polynomial degree k; see [3,5]. For boundary layers, the width of the thinnest layer mesh needs to be comparable to that of the boundary layer; see [36, 51, 52].

Geometric boundary layer mesh $\mathcal{T} = \mathcal{T}^{n,\sigma}_{bl}$ are obtained as tensor products of meshes that are geometrically refined towards the faces. The mesh $\mathcal{T}^{n,\sigma}_{bl}$ is built from an initial shape-regular macro-triangulation $\mathcal{T}^0$, possibly consisting of just one element, which is successively refined. Every macroelement can be refined isotropically or anisotropically as face, edge, or corner patch. A refinement towards a corner is shown in Figure 3. We refer the reader to [56, 58] for more details on the construction of these meshes. Note that the mesh aspect ratio is equal to $\sigma^{-n} \sim \sigma^{-k}$ since n needs to be comparable to k for exponential convergence.

A geometric boundary layer mesh $\mathcal{T}$ satisfies the following two properties:

Property 1.1. *$\mathcal{T}$ is obtained from an initial shape-regular coarse mesh $\mathcal{T}^0$ (macromesh) by local isotropic or anisotropic refinement.*

Property 1.2. *Anisotropic refinement is always performed towards the boundary $\partial\Omega$ of the computational domain Ω and never towards the interior.*

Figure 3 highlights these features. Both properties appear so far to be essential for the analysis; see [58] for more details. However good numerical results are obtained for more general situations; see Section 3.5.

2. Balancing Neumann-Neumann method

Given a geometric boundary layer mesh $\mathcal{T}$ and a spectral polynomial degree k, a function $u \in X^k(\Omega;\mathcal{T})$ is expanded using the basis functions described in Section 1.2. The finite element approximation of Problem (1) thus leads to a linear system

$$A\,u = b\,,$$

with A symmetric, positive-definite. The condition number of A can be huge for large values of k and n (see Sect. 3 for some numerical results) and efficient and robust preconditioners are therefore often mandatory. In this work, we investigate the Balancing Neumann-Neumann [35] iterative method. We refer the reader to [58] for a detailed derivation. More general information on domain decomposition methods can be found in the monographs [43, 53].

2.1. Subdomain partitions

Iterative substructuring methods rely on a non-overlapping partition of Ω, $\mathcal{T}^{DD} = \{\Omega_i\}$, into substructures. Let M denote the number of substructures with H_i the diameter of Ω_i and $H = \max(H_i)$ the maximum of their diameters. A subdomain Ω_i is called *floating* if the intersection of $\partial\Omega_i$ with $\partial\Omega$ is empty. We recall that we have only considered the case of Dirichlet boundary conditions. We define the boundaries $\Gamma_i = \partial\Omega_i \setminus \partial\Omega$ and the interface Γ as their union. The sets of Gauss-Lobatto nodes and the corresponding degrees of freedom on $\partial\Omega_i$, Γ_i, Γ, and $\partial\Omega$ are denoted by $\partial\Omega_{i,h}$, $\Gamma_{i,h}$, Γ_h, and $\partial\Omega_h$, respectively.

In this work, the main geometric assumption on the substructures is that they be *shape-regular*. This property appears to be essential to obtain the condition number bound presented in Section 2.4. Indeed, Property 1.1 allows us to fulfill this condition easily by choosing the macromesh as the subdomain partition:

$$\mathcal{T}^{DD} = \mathcal{T}^0.$$

A consequence of Property 1.2 is then that, when two substructures share an interior vertex, the local meshes are shape-regular in the neighborhood of this vertex, since anisotropic refinement is only performed towards the boundary $\partial\Omega$. This property also appears to be essential to obtain the condition number bound in Section 2.4.

2.2. Derivation

After subassembling, the stiffness matrix A is reordered according to the domain decomposition partitioning. The nodal points interior to the substructures (subset I) are ordered first, followed by those on the interface Γ (subset Γ). Similarly, for the local stiffness matrix relative to a substructure Ω_i, we have

$$A^{(i)} = \begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma}^{(i)} \\ A_{\Gamma I}^{(i)} & A_{\Gamma\Gamma}^{(i)} \end{pmatrix}.$$

First, the unknowns in the interior of the substructures are eliminated by block Gaussian elimination. Unknowns on $\partial\Omega_i \cap \partial\Omega$ are treated as interior and they are also eliminated. In this step, the Schur complement $S = S_{NN}$ with respect to the interior variables is formed. The resulting linear system for the nodal values on Γ can be written as

$$S_{NN}\,u_\Gamma = g_\Gamma. \tag{4}$$

Given the local Schur complement associated to the substructure Ω_i and the local right-hand side

$$S_i = A_{\Gamma\Gamma}^{(i)} - A_{\Gamma I}^{(i)} {A_{II}^{(i)}}^{-1} A_{I\Gamma}^{(i)} \qquad g_{\Gamma_i} = b_{\Gamma_i} - A_{\Gamma I}^{(i)} {A_{II}^{(i)}}^{-1} b_I^{(i)}, \tag{5}$$

the global Schur complement and the corresponding right-hand side g_Γ can be written as

$$S = S_{NN} = \sum_{i=1}^{M} R_i^T S_i R_i \qquad g_\Gamma = \sum_{i=1}^{M} R_i^T g_{\Gamma_i}\,, \tag{6}$$

where the restriction matrix R_i is a matrix of zeros and ones which extracts the variables on the local interface Γ_i from a vector of nodal values on Γ.

The Balancing Neumann-Neumann preconditioner $\hat{S}^{-1}$ [35] provides a preconditioned operator P_{NN} of the following form

$$P_{NN} = \hat{S}^{-1} S_{NN} = P_0 + (I - P_0)\left(\sum_{i=1}^{M} P_i\right)(I - P_0). \tag{7}$$

Here P_0 is associated to a low dimensional global coarse problem, whereas each operator P_i is associated to one substructure. More precisely, the local operators P_i are defined as:

$$P_i = R_i^T D_i S_i^\dagger D_i R_i S_{NN}, \tag{8}$$

where the matrices D_i are diagonal and $S_i^\dagger$ denotes either the inverse of S_i, if S_i is non-singular as for subdomains that touch $\partial\Omega$, or a pseudoinverse of S_i, if S_i is singular as for floating domains. In order to define the matrices $\{D_i\}$, we need to introduce a scaling function $\delta_i^\dagger$, which is a finite element function defined on the boundary $\partial\Omega_i$; *cf.* [16,17,35,41,48] . In order to define it it is enough to assign its values at the nodes in $\Gamma_{i,h}$. It is defined for $\gamma \in [1/2, \infty)$ and, is determined by a sum of contributions from Ω_i and its relevant next neighbors,

$$\delta_i^\dagger(x_l) = \frac{\left(a_{ll}^{(i)}\right)^\gamma}{\sum_{j\in\mathcal{N}_{x_l}} \left(a_{ll}^{(j)}\right)^\gamma}, \quad x_l \in \Gamma_{i,h}\,, \tag{9}$$

where $a_{ll}^{(i)}$ denotes the lth element of the diagonal of the local stiffness matrix $A^{(i)}$ and $\mathcal{N}_{x_l}$, $x_l \in \Gamma_h$, is the set of indices j of the subregions such that $x_l \in \Gamma_{j,h}$. We have chosen $\gamma = 1$ for our numerical experiments. Let D_i be the diagonal matrix with elements $\delta_i^\dagger(x)$ corresponding to the nodes in $\Gamma_{i,h}$.

The coarse space is defined as

$$V_0 = \text{span}\{R_i^T \delta_i^\dagger\},$$

where the span is taken over at least the floating subdomains. We denote by R_0^T the prolongation from the coarse to the global space. In analogy with (8), the coarse operator P_0 is defined as:

$$P_0 = R_0^T S_0^{-1} R_0 S_{NN}, \tag{10}$$

where $S_0 = R_0 S_{NN} R_0^T$ denotes the restriction of S_{NN} to that coarse space. We refer the reader to [58] for more details.

2.3. **Algorithm**

Since P_0 is a projection, a decomposition of the exact solution u of $P_{NN}u = \hat{S}^{-1}g_\Gamma$ can be found as

$$u = P_0 u + w, \qquad P_0 u = R_0^T S_0^{-1} R_0 g_\Gamma, \tag{11}$$

with $w \in \text{Range}(I - P_0)$. The Balancing Neumann-Neumann method reduces to a projected preconditioned Conjugate Gradient method in the space $\text{Range}(I - P_0)$ with an initial guess $u_0 = P_0 u + \tilde{w}$, with $\tilde{w} \in \text{Range}(I - P_0)$. The algorithm is given *e.g.* in ([59], Fig. 6.2). We remark that the matrices S_{NN} and $S_i^\dagger$ do not need to be calculated in practice. The action of S_{NN} on a vector requires the solution of a Dirichlet problem on

each substructure (application of the inverse of $A_{II}^{(i)}$), while the action of $S_i^\dagger$ can be calculated by applying a pseudoinverse of $A^{(i)}$ to a suitable vector, corresponding to the solution of a Neumann problem; see ([53], Chap. 4). One step of the algorithm therefore requires the solution of one Neumann and two Dirichlet problems on each substructure and one coarse problem ([59], Sect. 6.2.2).

2.4. **Condition number bound**

In the case of purely diffusive problems corresponding to $c = 0$ in (1), a bound for the condition number of the preconditioned operator P_{NN} restricted to the subspace Range$(I - P_0)$, to which the iterates are confined, has been proven in [58] for the case of exact solvers for Neumann and Dirichlet problems. We have:

$$\kappa(P_{NN}) \leq C\,(1-\sigma)^{-6}\,\left(1+\log\left(\frac{k}{1-\sigma}\right)\right)^2, \tag{12}$$

where the constant C is independent of the spectral polynomial degree k, the level of refinement n, the mesh grading factor σ, the coefficients ε and ρ, and the diameters of the substructures H_i. We note that $\kappa(P_{NN})$ does not depend on the number of substructures or the aspect ratio of the mesh and only depends polylogarithmically on the spectral polynomial degree k as in the p version on shape-regular meshes [41]. Finally, we remark that σ is bounded away from one and zero in practice, the optimal value being close to 0.15 [26].

2.5. **Inexact variant**

The global system reordered according to the domain decomposition partitioning can be written as

$$\hat{A}u = \hat{b} \quad \text{with} \quad \hat{A} = \begin{pmatrix} A_{II} & A_{I\Gamma} \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{pmatrix} \quad \hat{b} = \begin{pmatrix} b_I \\ b_\Gamma \end{pmatrix} \tag{13}$$

where $\hat{A}$ is symmetric, positive-definite. After a block Cholesky factorization of $\hat{A}$, its exact inverse can be written as

$$\hat{A}^{-1} = \begin{pmatrix} I & -A_{II}^{-1}A_{I\Gamma} \\ O & I \end{pmatrix} \begin{pmatrix} A_{II}^{-1} & O \\ O & S^{-1} \end{pmatrix} \begin{pmatrix} I & O \\ -A_{\Gamma I}A_{II}^{-1} & I \end{pmatrix}.$$

Thus given approximate solvers B_{II}^{-1} and B_S^{-1} for the interior and interface problems respectively, a preconditioner for $\hat{A}$ of the following form can be derived

$$M^{-1} = \begin{pmatrix} I & -B_{II}^{-1}A_{I\Gamma} \\ O & I \end{pmatrix} \begin{pmatrix} B_{II}^{-1} & O \\ O & B_S^{-1} \end{pmatrix} \begin{pmatrix} I & O \\ -A_{\Gamma I}B_{II}^{-1} & I \end{pmatrix}. \tag{14}$$

The inexact variant consists thus of solving the linear system (13) with the preconditioner (14) ([53], Sect. 4.4) or ([59], Sect. 4.3). The preconditioned operator is thus $P = M^{-1}\hat{A}$ and we will provide numerical estimates for the condition number $\kappa(P)$ in Section 3. The Balancing Neumann-Neumann procedure with inexact solvers for the Dirichlet and Neumann problems is used as an approximate solver for the interface problem, whereas approximate interior solvers are preconditioned conjugate gradient methods detailed later. Due to the choice of stopping criterion in the local solvers, the global preconditioner is found to be variable. We have therefore considered a Krylov subspace method [24] that accepts variable preconditioning when solving (13). Finally note that the global matrix-vector products of type $\hat{A}v$ where v is a vector of appropriate dimension are done exactly in the flexible inner-outer Krylov subspace method.

Local solvers. For the local solvers, preconditioned conjugate gradient solvers have been used here, which are stopped after a reduction of the initial residual of a factor of 10^{-4} or after 400 iteration steps. This type of stopping criterion involves thus a variable preconditioning strategy and requires to consider an appropriate outer solver as discussed before. In practice we have found numerically that far less than 400 iterations steps were needed to reach this reduction in most of the cases. This may be due to the quality of the preconditioner detailed next. However for certain applications this can be an expensive strategy in term of computational times. A cure will be proposed in Section 3.3.3.

Dirichlet and nonsingular Neumann local problems. In our numerical experiments, we have employed a conjugate gradient iteration with a factorized sparse approximate inverse (FSAI) as a preconditioner ([7], Sect. 5). We refer to [4] for a general introduction to sparse approximate inverse preconditioners. Given a symmetric positive definite matrix B, this approach based on an incomplete biconjugation process builds a factorized sparse approximate inverse of the form:

$$M = ZDZ^T \approx B^{-1} \tag{15}$$

where Z is a sparse unit upper triangular matrix and D is diagonal. Sparsity is preserved by dropping small elements in Z. A drop tolerance of 0.1 has been used here. Note that this construction is known to be breakdown-free for symmetric positive definite matrices [7]. As for incomplete factorization methods, the performance of approximate inverse preconditioners in factorized form is sensitive to the ordering of the matrix. As advised in [7], we have used a symmetric minimum degree reordering to reduce fill-in in the inverse factor Z and to improve the performance of the preconditioner. We refer to [12] for an effective construction of the preconditioner and the references quoted in ([7], Sect. 5.1.2), for extensive numerical experiments. To analyze the memory requirements needed by the FSAI preconditioner, we define its sparsity ratio τ as:

$$\tau = \frac{nnz(Z) + nnz(D)}{nnz(B)} \tag{16}$$

where $nnz(B)$ denotes the number of non-zero entries of matrix B. In our domain decomposition framework, we will derive the minimum and maximum values of the sparsity ratio over the subdomains for Dirichlet ($\tau^D_{\min}$, $\tau^D_{\max}$) and Neumann problems ($\tau^N_{\min}$, $\tau^N_{\max}$) respectively.

Neumann local problems on floating subdomains. For each floating subdomain (if any), the action of $S_i^\dagger$ can be calculated by applying a pseudoinverse of $A^{(i)}$ to a suitable vector ([53], Chap. 4). A pseudoinverse ${A^{(i)}}^\dagger$ can be explicitly computed by performing a singular value decomposition of $A^{(i)}$ and using the Moore-Penrose formulation [23]. Instead of performing any factorization, we propose a second route based on an iterative process. We consider the singular system of size $n \times n$:

$$A^{(i)}x = b \tag{17}$$

and recall that $A^{(i)}$ is symmetric and the index of the zero eigenvalue of $A^{(i)}$ is equal to 1. Note that due to the balancing procedure [35], this system is consistent *i.e.* $b \in \mathrm{Ker}(A^{(i)})^\perp = \mathrm{Range}(A^{(i)})$. Theorem 1.2.10 in [11] characterizes the pseudoinverse solution $x = {A^{(i)}}^\dagger b$ as the least-squares solution x of (17) (*i.e.* $x \in \mathbb{R}^n$ for which $\|b - A^{(i)}x\|_2$ is minimal) such that $\|x\|_2$ is minimal. We note that in our case the minimum of the residual is zero. Theorem 2 in [29] ensures that a solution to the singular system (17) can be found using a Krylov subspace method. We recall that GMRES [46] minimizes the same residual $\|b - A^{(i)}x\|_2$ over a Krylov subspace. Since system (17) is consistent and $\mathrm{Ker}(A^{(i)}) = \mathrm{Ker}({A^{(i)}}^T)$, then Theorem 2.4 in [13] assures that applying GMRES to (17) with $x_o \in \mathrm{Range}(A^{(i)})$ as starting vector will converge to the pseudoinverse solution. This procedure does not require any storage of factorization and is therefore well suited for large systems.

3. Numerical experiments

In this section, we provide a comparison between the condition number $\kappa(P)$ obtained numerically with the inexact variant of the preconditioner and the corresponding theoretical bound (12) obtained in case of exact solvers on small, medium and large size problems. We will devote a special attention to problems defined on highly stretched meshes, since they are generally challenging for iterative methods partly due to the bad condition numbers of the resulting matrices. In [58] a first small set of results has been already presented. A comparison between inexact and exact variants of the Balancing Neumann-Neumann preconditioner was provided for a Laplace problem on a boundary layer mesh (see Problem III in Sect. 3.3). Here we will only consider inexact solvers for the local Neumann and Dirichlet problems. For large size problems, this is a common practice to reduce computational costs by a significant amount.

We follow the same methodology as in our previous paper on numerical experiments for two-dimensional problems [57]. The first two test cases (Problems I, II) are recognized as standard test problems for domain decomposition preconditioners; see [35, 42, 53]. These problems defined on *shape-regular* and *uniform meshes* have been chosen here as a first evaluation step before tackling more involved problems. Problems III–VI are defined on *highly anisotropic meshes*. Whereas Problems III and V have been chosen as a natural extension of Problems I and II approximated on highly anisotropic meshes, Problem IV investigates the influence of the mesh grading factor σ on the condition number of the preconditioned operator. Problem VI investigates the behaviour of the domain decomposition preconditioner for a singularly perturbed problem of reaction-diffusion type.

It is known that p and hp-finite element discretizations lead to sparse matrices with more non-zero entries than classical h-finite element methods. In the tables we will report the size of the global matrix and the corresponding number of non-zero entries denoted by $size$ and nnz respectively. Using these quantities the number of operations involved in a global matrix vector product operation can be easily derived. In addition, the condition number estimates are obtained by computing the eigenvalues of the tridiagonal symmetric Lanczos matrix built in the conjugate gradient based outer process; see [6] for details. The minimum and maximum eigenvalues ($\lambda_{\min}$ and $\lambda_{\max}$, respectively) are also reported. The number of iterations It to reduce the Euclidean norm of the residual $\|r\|_2$ by twelve orders of magnitude

$$\|r_{It}\|_2/\|r_0\|_2 \leq 10^{-12} \tag{18}$$

is also reported. This rather strict stopping criterion allows a possible comparison between the proposed Balancing Neumann-Neumann preconditioner and preconditioners derived from the FETI framework [18] (since the primal solution in the one-level FETI formulation is only continuous at convergence). Memory requirements and solution time needed by the Balancing Neumann-Neumann preconditioner will be also partially detailed. Thus a comparison with direct solvers is also possible. As initial guess, a zero initial field is used for all problems.

As additional information, we have also reported the condition number of the global matrix A noted $\kappa(A)$ obtained by an iterative eigenvalue solver of Jacobi-Davidson type [20]. Finally for certain problems, we have reported the computation time in seconds (denoted by T) required to satisfy the reduction criterion (18).

To obtain the numerical results presented here, we have built a code using a mixed-language programming approach based on Python and C following [22]. Mesh generation, mesh decomposition, input and output routines, *e.g.*, are implemented in Python, whereas the time-critical tasks (mainly sparse algebra, iterative solvers and preconditioners) are implemented in C and integrated into Python. Indeed we have extended the library Pysparse developed by Roman Geus[1] to build our Python domain decomposition library. The main computations have been done on one processor of a 64-bit Sun machine with 32 Gb of main memory (CPU 900 Mhz).

[1] `http://people.web.psi.ch/geus/pyfemax/pysparse.html`

TABLE 1. Laplace problem. Domain decomposition preconditioned iterative method for the global system: size of the problem, number of non-zeros in the global matrix, iteration counts, maximum and minimum eigenvalues, condition numbers, and solution time in seconds versus polynomial degree and N, respectively. The total number of substructures is N^3.

Fixed number of substructures ($N = 8$)							
k	size	nnz	It	$\lambda_{\max}$	$\lambda_{\min}$	$\kappa(P)$	$T(s)$
2	4913	20051	10	1.3214	1.0000	1.3214	24
3	15625	84589	14	1.7511	1.0002	1.7508	76
4	35937	246495	17	2.2880	1.0001	2.2877	167
5	68921	574505	20	2.8559	1.0002	2.8553	326
6	117649	1200331	22	3.3573	1.0002	3.3565	575
7	185193	2326341	24	3.8386	1.0000	3.8388	988
8	274625	3778583	26	4.2805	1.0003	4.2794	1607
9	389017	12152929	28	4.7005	0.9998	4.7014	2651
10	531441	32192187	30	5.0922	0.9980	5.1021	5553
Fixed spectral degree $k = 10$							
N	size	nnz	It	$\lambda_{\max}$	$\lambda_{\min}$	$\kappa(P)$	$T(s)$
2	9261	411837	17	3.9177	0.9996	3.9193	31
3	29791	1522372	25	4.6553	0.9999	4.6557	184
4	68921	3771671	27	4.8795	0.9997	4.8808	494
5	132651	7561518	28	4.9787	0.9994	4.9815	1045
6	230731	13293697	29	5.0341	0.9960	5.0543	1921
7	363167	21369992	29	5.0689	0.9945	5.0970	3260
8	531441	32192187	30	5.0922	0.9980	5.1021	5553
9	753571	46162066	30	5.1087	0.9975	5.1217	8753
10	1030301	63681413	31	5.1209	0.9965	5.1388	18462
11	1367631	85152012	31	5.1303	0.9958	5.1518	28108

3.1. **Problem I: a Laplace problem**

We have first considered the Laplace operator with homogeneous Dirichlet boundary conditions:

$$\begin{aligned} -\Delta u &= 1, \quad \text{in } \Omega, \\ u &= 0, \quad \text{on } \partial\Omega. \end{aligned} \tag{19}$$

The unrefined $\mathcal{T}$ mesh is of Cartesian type and consists of N^3 cubes. Since in Problems I and II unrefined meshes are used, $\mathcal{T}^{DD}$ and $\mathcal{T}$ are identical. The total number of substructures is N^3.

The results are shown in Table 1. In the upper half of this table, the number of substructures is kept fixed ($N^3 = 8^3$), while the spectral polynomial degree k is varying from 2 to 10. In the lower half the spectral degree k is fixed to 10, while the number of substructures increases from 2^3 to 11^3. The first two columns report the size of the problem and the number of non-zeros in the global matrix. The next four columns report the iteration counts required to satisfy the stopping criterion (18), the maximum and minimum eigenvalues, and the condition number of the domain decomposition preconditioned operator.

The iteration count for the outer method appears to be bounded independently of N^3; see lower parts of Table 1. The condition numbers $\kappa(P)$ are plotted in Figure 1 *versus* the spectral polynomial degree k. As expected, the growth is quadratical in $\log(k)$.

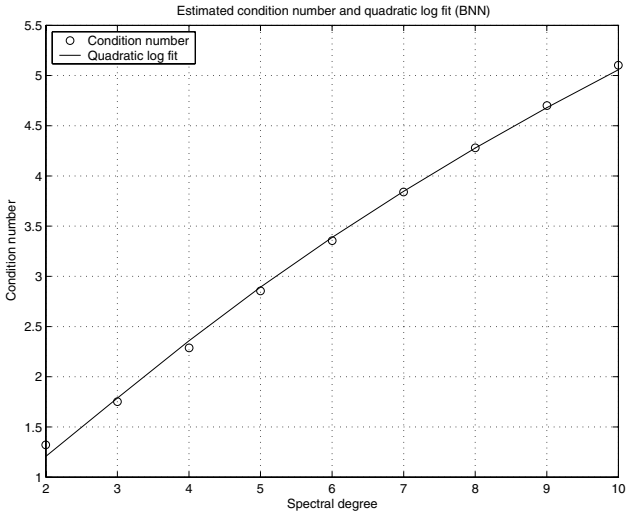


FIGURE 1. Laplace problem. Estimated condition numbers (circles) and least-square second order logarithmic polynomial fit (solid line) versus the spectral degree for the preconditioned operator (results from Table 1).

The case $k = 10$ and $N = 11$ is actually one of the largest problems we have solved and deserves some comments. For this case, the *hp* finite element discretization leads to a global matrix of size 1367631 with approximately 85.15 millions of non-zero entries. This requires a storage of 1.3 Gb only for this matrix. The total amount of storage for the factorized approximate inverses of $A_{II}^{(i)}$ (needed during the solution of Dirichlet problems) is about 73.5 Mb. Indeed the minimum and the maximum sparsity ratio are $\tau_{\min}^{D} = 0.0989$ and $\tau_{\max}^{D} = 0.1097$ respectively. On the other hand, the total amount of storage for the factorized approximate inverses of A_{II} (required for the solution of Neumann problems on non-floating subdomains) is about 62.8 Mb ($\tau_{\min}^{NN} = 0.1033$ and $\tau_{\max}^{NN} = 0.1194$). These values of sparsity ratio confirm that these FSAI preconditioners in general have low-memory requirements, while showing efficiency close to preconditioners coming from incomplete factorization methods. This was our main motivation for adopting them in this work. For the Neumann problems defined on floating subdomains, note that the computation of only one pseudoinverse is needed. Indeed the floating subdomains can only be internal ones where no refinement takes place. Since the same polynomial degree k is used throughout the mesh, only one pseudoinverse calculation is required. This induces an additional cost in storage of 27 Mb for this pseudoinverse. For this case, the ratio of the memory to store the domain decomposition preconditioner over the memory required to store the global matrix is about 0.2. As a consequence, the whole domain decomposition preconditioner is also attractive from a memory requirement point of view.

3.2. **Problem II: a Laplace problem with jump coefficients**

The theoretical bound for the condition number provided in (12) is independent of arbitrary jumps on the coefficients between the substructures. The purpose of this test case is to check this property. The following problem has been considered:

$$\begin{aligned} -\nabla \cdot (\rho \nabla u) &= 1, \quad \text{in } \Omega, \\ u &= 0, \quad \text{on } \partial\Omega, \end{aligned} \tag{20}$$

where the coefficient ρ possibly changes between the substructures by many orders of magnitude. Given a partition of $\Omega = (0,1)^3$ into N^3 cubic substructures ($\mathcal{T} = \mathcal{T}^0 = N \times N \times N$), a checkerboard distribution is considered for ρ which is equal to either ρ_1 or ρ_2 as in [35].

TABLE 2. Laplace problem with jump coefficients. Case of $\rho_1 = 10^{-3}$ and $\rho_2 = 10^3$. Domain decomposition preconditioned iterative method for the global system: size of the problem, number of non-zeros in the global matrix, condition number of the global matrix, iteration counts, maximum and minimum eigenvalues, and condition numbers versus polynomial degree. The total number of substructures is N^3.

Fixed number of substructures ($N = 3$)							
k	size	nnz	$\kappa(A)$	It	$\lambda_{\max}$	$\lambda_{\min}$	$\kappa(P)$
2	343	886	1.20E+06	6	1.2783	1.0000	1.2783
3	1000	3544	3.24E+06	10	1.7755	1.0000	1.7756
4	2197	10495	6.28E+06	12	2.0722	1.0000	2.0722
5	4096	25060	10.26E+06	12	2.3092	1.0000	2.3093
6	6859	53476	15.33E+06	14	2.4965	1.0000	2.4966
7	10648	105976	21.44E+06	15	2.6546	1.0000	2.6546
8	15625	174163	28.62E+06	15	2.7892	1.0000	2.7892
9	21952	571804	36.86E+06	16	2.9070	1.0000	2.9071
10	29791	1522372	46.21E+06	16	3.0113	1.0000	3.0114

3.2.1. *Fixed jumps between the substructures*

For a fixed partition into substructures with $N = 3$ and for fixed jumps between the substructures with $\rho_1 = 10^{-3}$ and $\rho_2 = 10^3$, we have investigated the behaviour of the condition number of the preconditioned operator versus the spectral polynomial degree k.

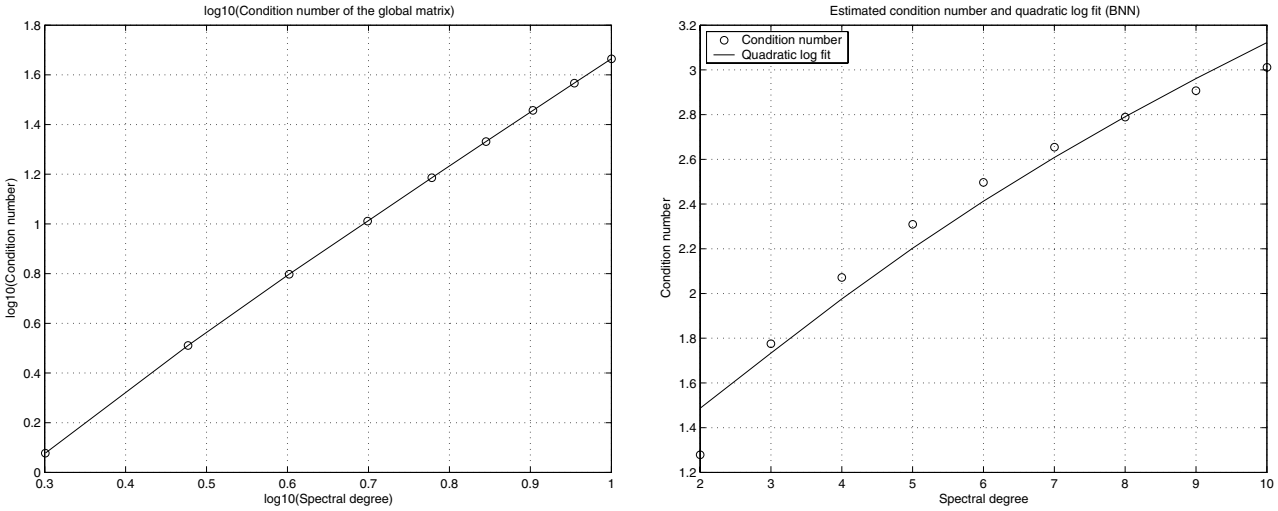
Table 2 shows the results. The behaviour of the condition number of the preconditioned operator is shown in Figure 2, right, and is consistent with the quadratic bound (12). We have also reported the condition number of the global matrix $\kappa(A)$ in Table 2; see also Figure 2 left. As expected a linear growth in $\log(k)$ is obtained for $\log(\kappa(A))$. We have found numerically that $\kappa(A)$ behaves as $k^{2.2}$ for this problem.

3.2.2. *Variable coefficient jumps*

Here the spectral polynomial degree k is fixed to 8. For four different partitions of type N^3 with $N = 2$, $N = 3$, $N = 4$ and $N = 5$, we have investigated the influence of the jump ρ_2/ρ_1 on the convergence behaviour of the preconditioned method. In this experiment, ρ_1 is fixed to 1, whereas ρ_2 is varying from 1 to 10^6. A checkerboard distribution has also been used.

The results are presented in Table 3. For each case, the size of the global matrix (size) and the number of non-zero entries (nnz) in this matrix are also reported. The number of preconditioned outer method iterations in order to satisfy the stopping criterion (18) is bounded independently of the ratio ρ_2/ρ_1, in agreement with the bound (12) for the case of exact solvers. For the solution of Neumann problems on floating domains (that exist here only for $N > 2$), we have employed the iterative version based on the preconditioned GMRES(m) solver described in Section 2.5. The size of the singular matrix is $(k + 1)^3$ with $k = 8$ in this case. As a preconditioner, the FSAI approach is used (see also [8] for using this kind of preconditioner in case of singular systems). We have observed that a low number of iterations is required to reach convergence. Finally note that the case ($N = 3$, $\rho_1 = 1, \rho_2 = 10^6$) of Table 3 is in good agreement with the result obtained for ($k = 8$, $\rho_1 = 10^{-3}, \rho_2 = 10^3$) in Table 2.

So far, we have only considered model problems on uniform meshes and shown that the numerical experiments are in agreement with the theoretical bound (12).



$log(k) - log(\kappa(A))$ k - $\kappa(P)$

FIGURE 2. Laplace problem with jump coefficients. Case of $\rho_1 = 10^{-3}$ and $\rho_2 = 10^3$. Fixed partition $3 \times 3 \times 3$. Dependence of the logarithm of the condition number of the global matrix on the logarithm of the spectral degree, left. Estimated condition numbers (circles) and least-square second order logarithmic polynomial (solid line) versus the spectral degree for the preconditioned method, right. Results from Table 2.

3.3. **Problem III: a Laplace problem on a boundary layer mesh**

Problem III is a Laplace problem with homogeneous Dirichlet boundary conditions defined on a boundary layer mesh.

$$\begin{aligned} -\Delta u &= 1, \quad \text{in } \Omega, \\ u &= 0, \quad \text{on } \partial\Omega. \end{aligned} \tag{21}$$

Geometric refinement is performed towards the corner located in $(0, 0, 0)$, with $\sigma = 0.5$ in each direction; see Figure 3. We note that this is a genuine *hp* approximation. As shown in [3, 5, 36, 51], in order to obtain exponential convergence in presence of singularities in polyhedral domains, the number of layers n must be at least equal to the spectral degree k, thus better accuracy is achieved by simultaneously increasing the polynomial degree and the number of layers. In our experiments we have chosen $n = k$.

3.3.1. *Fixed spectral degree*

Here the spectral polynomial degree k is fixed to 4. Given a uniform macromesh $\mathcal{T}^0$ of size N^3, we consider refinements with 4 layers in each direction (see Fig. 3, left). The non-uniform geometrically refined grid $\mathcal{T}$ contains $(N+4)^3$ elements (see Fig. 3, left, for a partition with $N = 5$), whereas the subdomain partition $\mathcal{T}^{DD}$ has N^3 substructures.

Table 4 shows the results for different partitions $\mathcal{T}^{DD}$ of type $N \times N \times N$. The iteration counts are uniformly bounded as the number of macroelements (subdomains) grow.

3.3.2. *Fixed partition*

We now fix a macromesh with $N = 3$ and investigate the dependence of the condition number $\kappa(P)$ on the spectral polynomial degree. The geometrically refined grid $\mathcal{T}$ contains $(3 + k)^3$ elements; see Figure 3, right, for the case $k = 6$. Table 5 shows the results. In the last column of this table referenced as *Exact*, we have reported – when available – estimates of the condition number of the Balancing Neumann-Neumann

TABLE 3. Laplace problem with jump coefficients. Case of $k = 8$ and $\rho_1 = 1$. Domain decomposition preconditioned iterative method for the global system: iteration counts, maximum and minimum eigenvalues, and condition numbers *versus* ρ_2. The total number of substructures is N^3.

	$N = 2$				$N = 3$			
	size $= 4913$		$nnz = 46031$		size $= 15625$		$nnz = 174163$	
ρ_2	It	$\lambda_{\max}$	$\lambda_{\min}$	$\kappa(P)$	It	$\lambda_{\max}$	$\lambda_{\min}$	$\kappa(P)$
1	15	3.2705	0.9995	3.2722	21	3.9101	1.0001	3.9098
10	13	2.9242	0.9994	2.9259	19	3.2379	1.0001	3.2377
10^2	12	2.6059	0.9996	2.6069	16	2.8014	1.0000	2.8015
10^3	11	2.5651	0.9997	2.5658	15	2.7475	1.0000	2.7475
10^4	11	2.5609	0.9997	2.5616	15	2.7420	1.0000	2.7420
10^5	11	2.5605	0.9997	2.5614	15	2.7891	1.0000	2.7891
10^6	12	2.5676	0.9997	2.5683	15	2.7892	1.0000	2.7892
	$N = 4$				$N = 5$			
	size $= 35937$		$nnz = 436119$		size $= 68921$		$nnz = 879719$	
ρ_2	It	$\lambda_{\max}$	$\lambda_{\min}$	$\kappa(P)$	It	$\lambda_{\max}$	$\lambda_{\min}$	$\kappa(P)$
1	23	4.1030	1.0001	4.1026	25	4.1874	1.0001	4.1868
10	21	3.4385	1.0001	3.4380	22	3.4420	1.0001	3.4415
10^2	19	3.0660	1.0000	3.0659	20	3.0671	1.0000	3.0671
10^3	18	3.0115	1.0000	3.0115	19	3.0029	1.0000	3.0031
10^4	18	3.0065	1.0000	3.0066	19	2.9994	1.0000	2.9995
10^5	18	2.9844	1.0000	2.9844	18	2.9899	1.0000	2.9900
10^6	18	3.0063	1.0000	3.0063	19	2.9947	0.9999	2.9949

TABLE 4. Laplace problem on a *boundary layer mesh* with $\sigma = 0.5$ and $n = 4$. Domain decomposition preconditioned iterative method for the global system: size of the global matrix, number of non-zero entries, iteration counts, maximum and minimum eigenvalues, condition number, and solution time in seconds versus the number of subdomains. The total number of substructures is N^3.

Fixed spectral degree $k = 4$							
N	size	nnz	It	$\lambda_{\max}$	$\lambda_{\min}$	$\kappa(P)$	$T(s)$
2	15625	98443	15	2.6415	0.9998	2.6421	47
3	24389	160679	21	3.9505	0.9999	3.9508	88
4	35937	244719	24	4.1081	0.9999	4.1084	130
5	50653	353827	24	4.1376	0.9999	4.1380	176
6	68921	491267	25	4.1490	0.9998	4.1497	244
7	91125	660303	26	4.1552	0.9997	4.1564	372
8	117649	864199	26	4.1305	0.9996	4.1321	553
9	148877	1106219	26	4.1609	0.9995	4.1629	822
10	185193	1389627	26	4.1633	0.9994	4.1659	1228

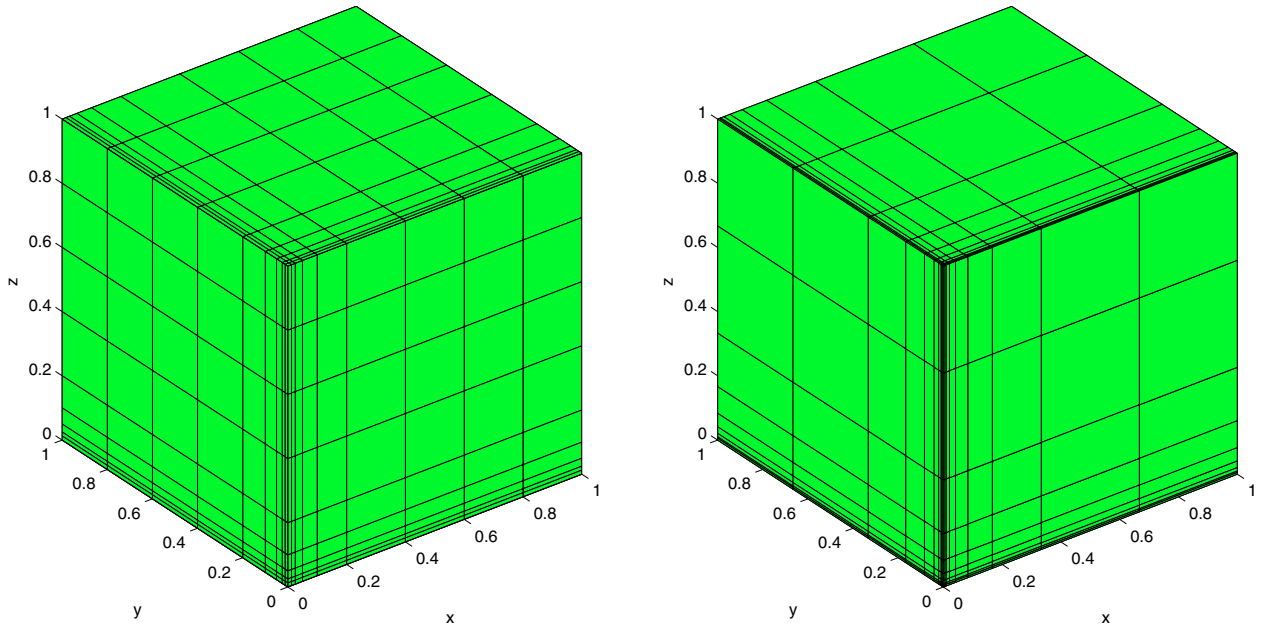


FIGURE 3. Geometric refinement towards one corner ($N = 5$, $\sigma = 0.5$, and $n = 4$), left and ($N = 3$, $\sigma = 0.5$, and $n = 6$), right.

TABLE 5. Laplace problem on a *boundary layer mesh* with $\sigma = 0.5$ and $n = k$. Domain decomposition preconditioned iterative method for the global system: size of the global matrix, number of non-zero entries, iteration counts, maximum and minimum eigenvalues, condition numbers, and solution time in seconds versus the polynomial degree. The total number of substructures is 3^3.

Fixed number of substructures ($N = 3$)								
			Inexact					Exact
k	size	nnz	It	$\lambda_{\max}$	$\lambda_{\min}$	$\kappa(P)$	$T(s)$	$\kappa(P_{NN})$
2	1331	4.3E+03	13	1.8379	1.0000	1.8379	2.00E+00	1.6255
3	6858	3.3E+04	17	2.8163	1.0000	2.8163	1.20E+01	2.8161
4	24389	1.6E+05	21	3.9505	0.9999	3.9507	8.80E+01	3.9498
5	68921	5.7E+05	25	5.1505	0.9997	5.1518	7.85E+02	5.1493
6	166375	1.7E+06	29	6.3675	9.9916	6.3728	4.87E+03	6.3658
7	357911	4.6E+06	32	7.5067	0.9984	7.5184	2.03E+04	7.5065
8	704969	1.0E+07	34	8.5294	0.9973	8.5525	6.11E+04	8.5062
9	1297645	4.2E+07	36	9.4229	0.9969	9.4520	2.84E+05	-
10	2251235	1.4E+08	38	10.267	0.9967	10.301	1.30E+06	-

preconditioner $\kappa(P_{NN})$, when an exact variant of the preconditioner was used. These results for the exact variant have been already presented (see Tab. 2 in [58]).

Figure 4 shows the behaviour of the condition number versus the polynomial degree. As expected a quadratical growth in $\log(k)$ is obtained. In a previous paper (see Tab. 2 in [58]), we have made a similar analysis on the same problem and compared inexact and exact variants of the domain decomposition preconditioner.

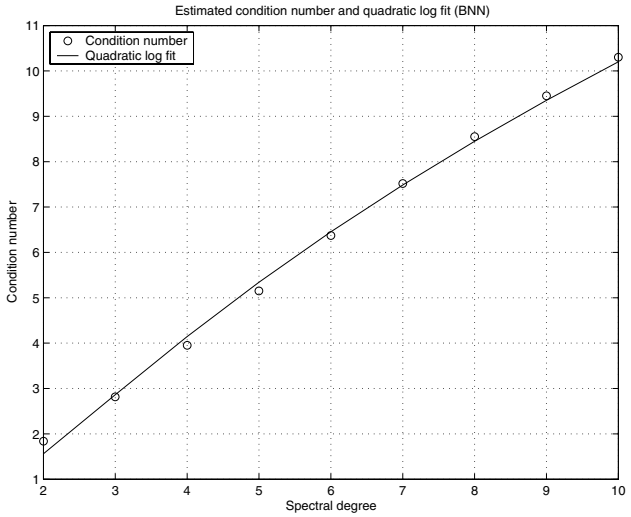


FIGURE 4. Laplace problem on a boundary layer mesh. Estimated condition numbers (circles) and least-square second order logarithmic polynomial (solid line) versus the spectral degree for the preconditioned operator (results from Tab. 5).

An exact variant was derived, when solving all the local subproblems now up to machine precision with the same iterative solver as in the inexact case. Condition numbers obtained in [58] for the inexact variant are recalled in Table 7 (case $\sigma = 0.5$). Note that the stopping criterion that was used in [58] was *fourteen* orders of magnitude of residual reduction explaining the difference in iteration counts between Table 5 and Table 2 in [58]. As expected the condition numbers are in good agreement. The slight differences on the condition numbers between both inexact variants are due to a different stopping criterion when solving inexactly Neumann and Dirichlet problems. The condition numbers shown in Table 5 are closer to the ones of the exact variant. This is mainly due to the fact that we have solved both Neumann and Dirichlet local problems more accurately than in [58]. These improvements appear only for k large, *i.e.*, as the size of the problem becomes larger and larger.

The case $k = 10$ leads to a matrix of size 2251235 with approximately 141.5 million non-zero entries. This requires a storage of 2.1 Gb. The total amount of storage for the factorized approximate inverses needed during the solution of Dirichlet and Neumann problems is about 3.9 Gb. For this case, the ratio of the memory to store the domain decomposition preconditioner over the memory required to store the global matrix is about 1.87. Note the difference with respect to Problem I. However this ratio is still acceptable, since in practice it should not exceed 2 or 3 as suggested in [47].

3.3.3. *Computational times and improvements*

We note that very large computational times are obtained for problems defined on highly refined meshes ($k \geq 7$) in Table 5. Indeed the sizes of the Neumann local problems for the subdomain that touches the corner $(0, 0, 0)$ are 185193, 389017, 735571 and 1367631 for $k = 7, 8, 9, 10$ respectively. For, *e.g.*, $k = 10$ this means that this subdomain contains 60 percent of the total number of degrees of freedom. Moreover these local problems are also expected to be very ill conditioned due to the mesh refinement procedure and the preconditioned conjugate gradient solver for the solutions of local Dirichlet and Neumann problems often requires all of the 400 iterations that we require for the stopping criterion; see Section 2.5. More efficient and robust local Dirichlet and Neumann solvers are therefore needed. Multigrid and/or iterative solvers exploiting the tensor product type of the geometry should be considered as fast solvers for this subdomain to simultaneously improve the convergence rate of the local solver and decrease the computational cost. Once this efficient local solver is found, the next step is to consider parallelization to decrease substantially the total solution time.

TABLE 6. Laplace problem on a *boundary layer mesh* with $\sigma = 0.5$ and $n = k$. New strategy resulting in considerable gains in term of CPU time (same legend as in Tab. 5, here T_{ref} stands for the CPU time shown in Tab. 5).

Fixed number of substructures ($N = 3$)								
			Inexact					Gain
k	size	nnz	It	$\lambda_{\max}$	$\lambda_{\min}$	$\kappa(P)$	$T(s)$	T_{ref}/T
2	1331	4.3E+03	11	1.6255	0.9999	1.6256	1.54	1.299
3	6858	3.3E+04	17	2.8097	0.9993	2.8114	7.80	1.792
4	24389	1.6E+05	22	3.0985	0.8723	3.5521	30.40	2.881
5	68921	5.7E+05	27	3.5098	0.6237	5.6272	121.70	6.452
6	166375	1.7E+06	33	3.8164	0.4700	8.1200	478.04	10.192
7	357911	4.6E+06	40	4.5206	0.3577	12.636	1557.94	13.017
8	704969	1.0E+07	49	4.7185	0.2880	16.381	4453.16	13.718
9	1297645	4.2E+07	62	5.1021	0.2012	25.258	20397.94	13.924

Here we aim at reducing the computational times presented in Table 5 (now referenced as T_{ref}) by requiring less precision when solving approximately local subproblems and consequently doing more outer iterations. We adopt a simple strategy by choosing two new ingredients. First we modify the stopping criterion for the inner solvers as follows: a fixed number of inner iterations (15) is required. Secondly we use a different preconditioner than that described in Section 2.5 when solving local Dirichlet or Neumann problems. This preconditioner still comes from sparse approximate inverse techniques but it is based now on Frobenius norm minimization [25]. An approximate inverse is derived by minimizing the functional $F(M) = \|I - BM\|_F$ subject to some sparsity constraints, where $\|.\|_F$ denotes the Frobenius norm; see also [4,7] for general comments. Instead of a dynamical approach that may be very costly, we fix *a priori* the sparsity constraints [15]. We select the pattern of the sparse approximate inverse as the pattern of the matrix that retains only the strong connections in B (as done in algebraic multigrid when coarsening [44]). Since the SPAI approach leads to a non symmetric preconditioner, we have considered the GMRES method [46] as inner solver. As outer solver, the flexible variant of GMRES [45] has been considered. Results are shown in Table 6. In this table we have reported in the last column the ratio between the computational times in Table 5 and those of this modified algorithm (Gain). A considerable decrease in time is observed and thus attractive gains are obtained despite this rather simple approach. We recall that the original system is solved with a precision of 10^{-12}. In practice smaller precisions are generally employed: if we choose a value of 10^{-6} for the stopping criterion (18), we expect a reduction of computational times by a factor of one half [4]. This means a computational time of approximately 37 minutes on a 900 Mhz processor for the case $k = 8$ in Table 6. This is a reasonable result for this size of problem (704969 unknowns) and for the large aspect ratio. For larger problems however a parallel implementation should be considered. Approximately solving local problems with the new strategy leads as expected to an increase in iteration counts. Note that the minimum eigenvalue is no any longer close to 1.

3.4. **Problem IV: influence of the mesh grading factor**

In Problem III the mesh grading factor σ was kept fixed to 0.5 in each direction. In Problem IV we want to investigate the performances of the preconditioner with respect to the mesh aspect ratio. The mesh grading factor will now be variable and we only fix the total number of substructures. The polynomial degree k will also vary and as already explained in Problem III, the number of layers n will be equal to k. The aspect ratio is indeed influenced by n and σ. The same problem as in the previous Section 21 is solved. The numerical experiments of Problem IV have been partly carried out in Matlab 6.1. Here we only present results for corner refinement as in Problem III. The case of face and edge refinements shows a similar behaviour.

TABLE 7. Refinement towards a corner. Domain decomposition preconditioned iterative method for the global system: mesh aspect ratio (AS) *i.e.* σ^{-k}, iteration counts, and condition numbers versus the mesh grading factors and k. The total number of substructures is 3^3.

Refinement towards a corner ($3 \times 3 \times 3$ partition)						
σ	AS	It	$\kappa(P)$	AS	It	$\kappa(P)$
	$k=2$			$k=3$		
0.50	4.00E+00	15	1.8379	8.00E+00	20	2.8166
0.20	2.50E+01	15	2.0794	1.25E+02	21	3.2105
0.15	4.44E+01	15	2.1333	2.96E+02	21	3.2252
0.10	1.00E+02	15	2.1779	1.00E+03	21	3.2153
0.05	4.00E+02	15	2.2029	8.00E+03	20	3.1879
0.01	1.00E+04	15	2.2201	1.00E+06	20	3.1607
	$k=4$			$k=5$		
0.50	1.60E+01	25	3.9528	3.20E+01	29	5.1611
0.20	6.25E+02	26	4.3497	3.12E+03	29	5.3125
0.15	1.97E+03	26	4.3150	1.31E+04	29	5.2468
0.10	1.00E+04	25	4.2698	1.00E+05	28	5.1856
0.05	1.60E+05	25	4.2238	3.20E+06	28	5.1319
0.01	1.00E+08	25	4.1895	1.00E+10	28	5.0940
	$k=6$			$k=7$		
0.50	6.40E+01	34	6.3803	1.28E+02	38	7.5540
0.20	1.56E+04	33	6.1795	7.81E+05	35	6.9603
0.15	8.77E+04	32	6.0967	5.85E+05	35	6.8663
0.10	1.00E+06	31	6.0269	1.00E+07	34	6.7857
0.05	6.40E+07	31	5.9686	1.28E+09	34	6.7250
0.01	1.00E+12	31	6.0716	1.00E+14	35	6.6808

Given a partition of $\Omega = (0,1)^3$ into N^3 cubic substructures ($\mathcal{T} = \mathcal{T}^0 = N \times N \times N$) with $N = 3$, we have considered a geometrical refinement towards the corner located at $(0,0,0)$. This kind of refined mesh has already been used in Problem III (see Fig. 3).

Table 7 summarizes the results for various mesh grading factors (0.5, 0.2, 0.15, 0.1, 0.05 and 0.01). Note that to limit computational times and memory requirements, we have only considered a polynomial degree k varying from 2 to 7. However this feature does not restrict our analysis. *hp* approximations performed on highly stretched meshes do lead to huge condition numbers for the original global matrix A as shown in Figure 5, left, where *e.g.* $\kappa(A) = 9.07\,e+19$ for $\sigma = 0.05$ and $k = 7$. As expected an exponential growth of $\kappa(A)$ versus k is obtained. This is also confirmed in Figure 5, right, where $\kappa(A)$ is plotted versus the mesh aspect ratio in a log-log scale for a fixed spectral degree $k = 4$.

Figure 6, right, shows the condition number of the domain decomposition preconditioned operator versus the polynomial degree for various mesh grading factors (0.2, 0.15, 0.1 and 0.01). As expected, a quadratical growth in $\log(k)$ is obtained whatever the choice of the mesh grading factor. Note also that the number of iterations of the preconditioned outer method is practically independent of the mesh grading factor σ as shown in Figure 6, left. Moreover the preconditioner performs well, even if the mesh aspect ratio is huge (see Tab. 7, case $k = 7$ and $\sigma = 0.01$ leading to a mesh aspect ratio of 10^{14} and Fig. 6, left). This is a very attractive feature when solving problems where boundary layers and/or singularities occur. Note that the theory [37,50] asserts that the *hp* finite element method with *any* geometrically graded mesh gives exponential convergence. A careful choice of σ can result in discretization errors which are several orders of magnitude smaller than in the case $\sigma = 0.5$,

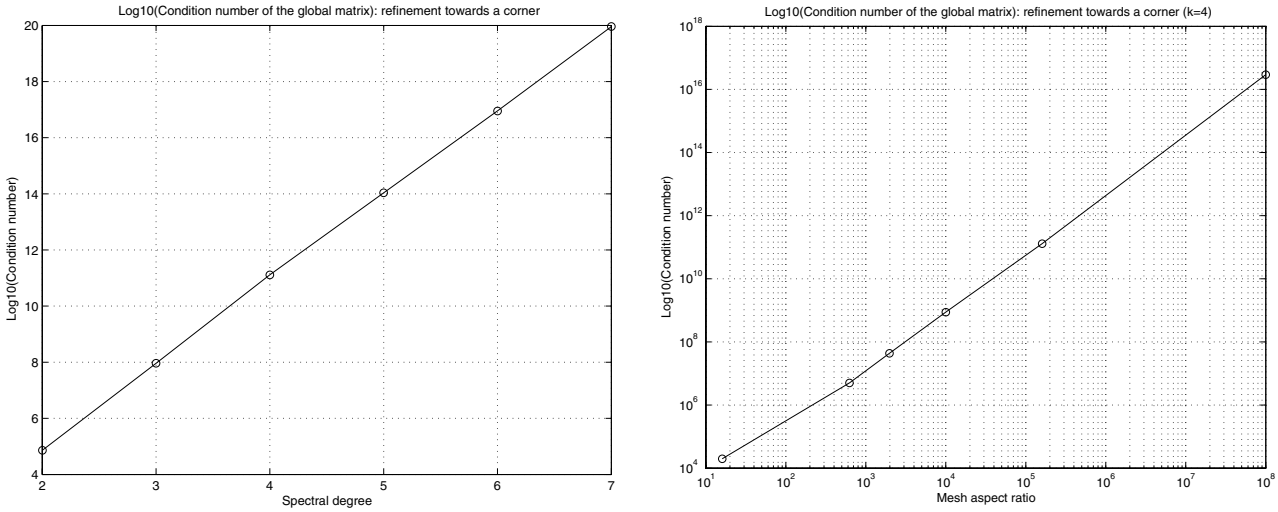


FIGURE 5. Refinement towards a corner. Dependence of the condition number of the global matrix $\kappa(A)$ on the spectral degree (left, $\sigma = 0.05$). Dependence of the condition number of the global matrix $\kappa(A)$ on the mesh aspect ratio σ^{-k} (log-log scale, right, $k = 4$).

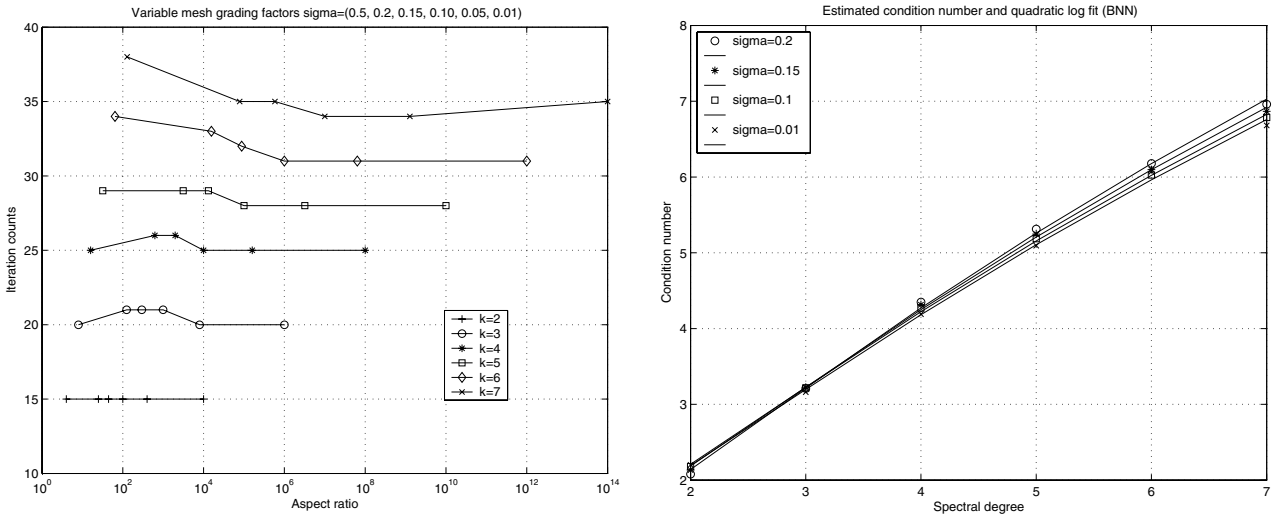


FIGURE 6. Refinement towards a corner. Iteration counts *versus* the mesh aspect ratio *i.e.* σ^{-k} for different mesh grading factors (left). Estimated condition numbers (circles) and least-square second order logarithmic polynomial (solid line) *versus* the spectral degree for the preconditioned operator for different mesh grading factors (right). Results from Table 7.

even though the number of degrees of freedom is the same. Thus in one dimension the optimal mesh grading factor is $\sigma^\star \approx 0.15$; see [26] or ([52], p. 96). Although the optimal σ is not explicitly known in two or three dimensions, we expect it to be of approximately the same order. We refer to [49] for a numerical illustration in two dimensions. Table 7 shows that the preconditioner performs well for mesh grading factors close to $\sigma^\star$.

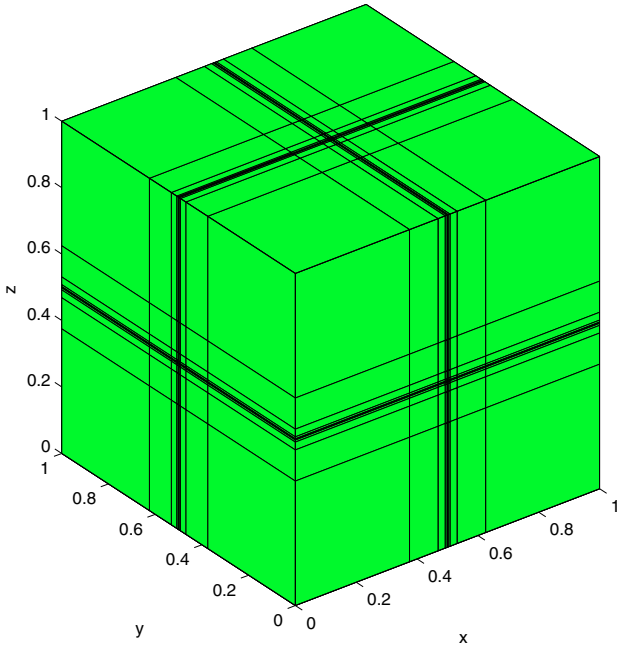


FIGURE 7. Interface problem. Anisotropic mesh with interior refinement for the case $n = k = 4$. Mesh grading factor $\sigma = 0.25$ towards the interface in each direction.

3.5. **Problem V: an interface problem**

Singularities may sometimes occur not only in the neighborhood of boundaries of polyhedral domains, as is investigated in Problems III and IV, but also at the interfaces of regions with different material properties. For example, interface problems in hydrology, reservoir modelling or nuclear waste management may require highly refined meshes inside the computational domain. Such problems, involving simultaneously jump coefficients and large aspect ratios of the mesh, are extremely important in practice. Our interface problem is defined as in Problem II:

$$\begin{array}{ll} -\nabla \cdot (\rho \nabla u) = 1, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega. \end{array} \tag{22}$$

We assume now that Ω is divided into eight equal cubes (see Fig. 7) and that the coefficient ρ has a checkerboard distribution, given by $\rho_1 = 10^4$ and $\rho_2 = 1$. The interface is thus made of the planes $x = 1/2$ $y = 1/2$ and $z = 1/2$. As a subdomain partition $\mathcal{T}^{DD}$ we choose that given by ρ. We then have $2 \times 2 \times 2$ substructures. In order to capture the interface effects, we have employed a geometrically refined mesh towards both sides of the interface. Since the purpose of this test case is to assess the properties of our preconditioners if anisotropic refinement takes place in the interior of Ω, we have neglected the effects of the singularities at $\partial\Omega$. Figure 7 shows the refined mesh $\mathcal{T}$. As in Problem IV, the number of layers is determined only by the spectral polynomial degree k. Thus the highly refined mesh $\mathcal{T}$ consists of $(2+2k)^3$ hexahedrals, thus providing an hp approximation of this problem. Mesh grading factors $\sigma_x = 0.25$, $\sigma_y = 0.25$ and $\sigma_z = 0.25$ towards the interface have been considered in this numerical experiment.

The spectral polynomial degree k is varying from 2 to 5. Table 8 shows the results. As in Problem IV note that the condition number $\kappa(A)$ of the global matrix is remarkably large already for small values of k and consequently moderate values of the mesh aspect ratio. From the results of Table 8, it can be checked that the growth of $\kappa(A)$ *versus* the spectral polynomial degree is exponential. Robust preconditioners are therefore mandatory for this kind of applications. As expected the obtained results lead to a different behaviour for the

TABLE 8. Interface problem. Domain decomposition preconditioned iterative method for the global system: size of the global matrix, number of non-zero entries, condition number of the global matrix, iteration counts, maximum and minimum eigenvalues, and condition numbers versus the spectral polynomial degree. The total number of substructures is 2^3.

Fixed number of substructures ($N = 2$)							
k	size	nnz	$\kappa(A)$	It	$\lambda_{\max}$	$\lambda_{\min}$	$\kappa(P)$
2	2197	7429	2.9317E+06	5	1.0001	0.9998	1.002
3	15625	82573	1.3476E+08	7	3.8221	0.9999	3.8221
4	69407	489203	5.5162E+09	8	4.6456	0.9999	4.6456
5	227707	2003017	1.5236E+11	9	5.4089	0.9999	5.4089

condition number of the preconditioned operator. Indeed Property 1.2 is not satisfied and the bound (12) for the condition number of the preconditioned operator is no more valid. Plotting $\kappa(P)$ versus the spectral degree k in a log-log plot does lead to a linear behaviour, in contrast to all the other numerical experiments shown in this paper. Numerically $\log(\kappa(P_{NN}))$ is found to grow like k^p with $p = 0.68$. Following Section 6 in [57] we expected a quadratic bound for the exact variant

$$\kappa(P_{NN}) \leq C \left(1 + \log\left(\frac{kH}{h}\right)\right)^2 \sim C \left(1 + \log\left(k\sigma^{-n}\right)\right)^2 \leq C\, k^2,$$

with $n \sim k$. Indeed, the results in Section 3.5 are consistent with a *linear* growth in k. Nevertheless this behaviour is already an improvement with respect to the exponential growth of the condition number of the original problem. Note also that an extremely small number of iterations is obtained to satisfy the convergence criterion (18) and the constant C appears to be very small in this case. The same behaviour has been also obtained on a two-dimensional version of this problem [57]. As a further study it would be interesting to perform some numerical experiments with other efficient multilevel preconditioners based either on algebraic multigrid [44, 60] or on algebraic domain decomposition ideas [47] among others. An efficient iterative solver may result in combining these techniques with deflation [39].

3.6. **Problem VI: a reaction-diffusion problem**

So far we have only considered purely diffusive problems. We note that the analysis provided in [56, 58] does not cover the case of reaction-diffusion problems. However for two-dimensional problems, numerical experiments [56, 57] have confirmed that the growth of the condition number of the preconditioned operator was also quadratic in $\log(k)$ for this type of problems. Thus we finally consider the following reaction-diffusion problem where ε is a possibly small real coefficient:

$$\begin{aligned} -\varepsilon\, \nabla \cdot (\nabla u) + u &= 1, && \text{in } \Omega, \\ u &= 0, && \text{on } \partial\Omega. \end{aligned} \tag{23}$$

The source term is not compatible with the boundary conditions and thus boundary layers appear for ε small. Geometrically refined meshes are then needed in order to achieve exponential convergence and robustness with respect to ε; see, *e.g.*, [36, 52]. Our main goal is here to analyse the convergence behaviour of the preconditioner for different values of ε.

Since boundary layer effects are present, the size of the thinnest layer $H\sigma^k$ should be comparable to the size of the boundary layer $\sqrt{\varepsilon}$; see [36, 52]. In addition, singularity resolution requires that n be comparable to k. These assumptions lead to the following relation to determine the level of refinement n and the spectral

TABLE 9. Reaction-diffusion problem on a boundary layer mesh. Domain decomposition preconditioned iterative method for the global system: spectral polynomial degree, size of the global matrix, number of non-zero entries, iteration counts, maximum and minimum eigenvalues, and condition numbers versus ε. The total number of substructures is 3^3.

Fixed number of substructures ($N = 3$)							
ε	k	size	nnz	It	$\lambda_{\max}$	$\lambda_{\min}$	$\kappa(P)$
1	2	1331	4268	9	1.2481	1	1.2481
10^{-1}	2	1331	4268	6	1.0446	1	1.0446
10^{-2}	2	1331	4268	4	1.0017	1	1.0017
10^{-3}	4	24389	160679	4	1.0043	1	1.0043
10^{-4}	6	166375	1723426	4	1.0084	1	1.0084
10^{-5}	7	357911	4614893	4	1.0017	0.9999	1.0017
10^{-6}	9	1297645	41924491	4	1.0011	0.9999	1.0011

polynomial degree k when $\varepsilon < 1$:

$$n = n(\varepsilon) = \left[\frac{\log(\sqrt{\varepsilon}/H)}{\log \sigma}\right] + 1, \quad k = k(\varepsilon) = n(\varepsilon),$$

where $[x]$ denotes the integer part of x. For $\varepsilon = 1$, refinement - although not needed - is performed ($n = 2$) and the spectral polynomial degree k is fixed to 2. The macromesh $\mathcal{T}^{DD}$ consists of 3^3 substructures ($H = 1/3$). Geometric refinement is only performed towards the corner located in $(0, 0, 0)$, with a mesh grading factor $\sigma = 0.5$ as in Problem III. The refined grid $\mathcal{T}$ contains thus $(3 + k)^3$ elements. We stress the fact that ε determines both n and k, and that we have here a genuinely hp approximation.

Table 9 shows the results for the domain decomposition preconditioner. We note that for $\varepsilon = 0$ the stiffness matrix A reduces to the mass matrix but mass matrices arising from spectral elements are not necessarily uniformly well-conditioned with respect to k even for shape-regular meshes. For one single spectral element, their condition number is expected to grow as k^3 for three-dimensional problems; see [56]. The domain decomposition preconditioner leads to very satisfactory results and the convergence behaviour is thus robust with respect to ε as well.

4. PERSPECTIVES

Many important issues remain to be partially or fully addressed:

As shown in the numerical experiments, hp finite element discretizations on three-dimensional boundary layer meshes can lead to large problems with more than two millions of degrees of freedom, even if the number of elements is moderate. Thus in order to reduce computational times, parallelization should be considered. In the domain decomposition framework, parallelization is generally achieved by using a domain partition of the refined mesh assigning to each processor one subdomain such that each processor contains approximately the same number of unknowns. Thus a well-balanced computation is expected. For Problems I, II and V, this strategy leads to use the macro-mesh $\mathcal{T}^0$ as domain partition for the parallel computations. However this choice is clearly inappropriate for *e.g.* Problem III (corner refinement see Fig. 3, right), since the subdomain that touches the corner $(0, 0, 0)$ will contain much more unknowns than the other subdomains (see the comments in Sect. 3.4). Thus one possible idea is to use a "macro-macro-mesh" deduced from $\mathcal{T}^0$ to equilibrate the load balancing. This simply means that possibly two or more subdomains could be assigned to one processor. Given this suitable subdomain to processor mapping, standard parallelization techniques by message passing can be used to build the parallel solver. Note that the parallelization of the FSAI or SPAI solvers for the Dirichet and

Neumann local problems is straightforward since it involves only matrix-vector products. The construction of the approximate solver is also parallelizable [7, 25].

When solving (possibly time-dependent) problems that may mix difficulties (corner singularities, boundary layer effects), *hp* adaptivity is required [14,28]. The elements can be subsected (isotropically and anisotropically) and their orders can be enriched which permits non-uniform distribution of element sizes h and orders p. Thus anisotropic polynomial degrees can be chosen in the computational domain, whereas h-refinement may lead to non-conforming meshes with hanging nodes that involve fewer degrees of freedom and thus gives smaller algebraic linear systems to solve than the ones we have obtained. We refer to [21] for implementation issues of *hp* adaptivity. Therefore one crucial issue to be addressed is the treatment of hanging nodes in the domain decomposition framework. We refer to [56, 58] for more comments.

The estimates derived in [58] can be employed to prove condition number bounds for certain type of FETI methods. It would be also interesting to analyze numerically the behaviour of the equivalent FETI preconditioner on the problems described in this paper. On two-dimensional problems this analysis has been conducted in [57] for one-level FETI. Another topic of research is to apply this analysis to convection-dominated convection-diffusion problems. We note that the preconditioners proposed in [1, 55] can be applied as well.

Finally in this work we have considered iterative solvers based on domain decomposition ideas. It would be interesting to perform a comparison in terms of computational efficiency and robustness with respect to the mesh aspect ratio with other solvers based either on multilevel ideas or fast solvers exploiting the tensor product type of the mesh. As investigated in recent studies [10,32,34], a clever combination of all these techniques could lead to an efficient and robust solver for *hp* finite element approximations.

References

[1] Y. Achdou, P. Le Tallec, F. Nataf and M. Vidrascu, A domain decomposition preconditioner for an advection-diffusion problem. *Comput. Methods Appl Mech. Engrg.* **184** (2000) 145–170.

[2] M. Ainsworth, A preconditioner based on domain decomposition for *hp*–FE approximation on quasi–uniform meshes. *SIAM J. Numer. Anal.* **33** (1996) 1358–1376.

[3] B. Andersson, U. Falk, I. Babuška and T. von Petersdorff, Reliable stress and fracture mechanics analysis of complex aircraft components using a *hp*–version FEM. *Int. J. Numer. Meth. Eng.* **38** (1995) 2135–2163.

[4] O. Axelsson, *Iterative Solution Methods*. Cambridge University Press (1994).

[5] I. Babuška and B. Guo, Approximation properties of the *hp*–version of the finite element method. *Comput. Methods Appl. Mech. Engrg.* **133** (1996) 319–346.

[6] R. Barrett, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine and H. Van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, 2nd edition. SIAM, Philadelphia, PA (1994).

[7] M. Benzi, Preconditioning techniques for large linear systems: a survey. *J. Comput. Phys.* **182** (2002) 418–477.

[8] M. Benzi and M. Tuma, A parallel solver for large-scale Markov chains. *Appl. Numer. Math.* **41** (2002) 135–153.

[9] C. Bernardi and Y. Maday, Spectral methods. In *Handbook of Numerical Analysis*, North-Holland, Amsterdam Vol. V, Part 2 (1997) 209–485.

[10] S. Beuchler, Multigrid solver for the inner problem in domain decomposition methods for *p*-fem. *SIAM J. Numer. Anal.* **40** (2002) 928–944.

[11] A. Björck, *Numerical methods for least-squares problems*. SIAM (1996).

[12] R. Bridson and W.-P. Tang, Refining an approximate inverse. *J. Comput. Appl. Math.* **123** (2000) 293–306.

[13] P. Brown and H. Walker, GMRES on (nearly) singular systems. *SIAM J. Matrix Anal. Appl.* **18** (1997) 37–51.

[14] W. Cecot, W. Rachowicz and L. Demkowicz, An *hp*-adaptive finite element method for electromagnetics. III. a three-dimensional infinite element for Maxwell's equations. *Internat. J. Numer. Methods Engrg.* **57** (2003) 899–921.

[15] E. Chow, *A priori* sparsity patterns for parallel sparse approximate inverse preconditioners. *SIAM J. Sci. Comput.* **21** (2000) 1804–1822.

[16] M. Dryja and O.B. Widlund, Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems. *Comm. Pure Appl. Math.* **48** (1995) 121–155.

[17] M. Dryja, M.V. Sarkis and O.B. Widlund, Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimensions. *Numer. Math.* **72** (1996) 313–348.

[18] C. Farhat and F.-X. Roux, Implicit parallel processing in structural mechanics, in *Computational Mechanics Advances*, J. Tinsley Oden Ed. North-Holland **2** (1994) 1–124.

[19] C. Farhat and F.-X. Roux, A method of finite element tearing and interconnecting and its parallel solution algorithm. *Int. J. Numer. Meth. Engng.* **32** (1991) 1205–1227.
[20] D.R. Fokkema, G.L.G. Sleijpen and H.A. Van der Vorst, Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils. *SIAM J. Sci. Comput.* **20** (1998) 94–125.
[21] P. Frauenfelder and C. Lage, An object oriented software package for partial differential equations. *ESAIM: M2AN* **36** (2002) 937–951.
[22] R. Geus, *The Jacobi-Davidson algorithm for solving large sparse symmetric eigenvalue problems with application to the design of accelerator cavities*. Ph.D. thesis, ETH, Zürich, Institut für Wissenschaftliches Rechnen (2002).
[23] G. Golub and C. Van Loan, *Matrix Computations*. The John Hopkins University Press (1996). Third edition.
[24] G. Golub and Q. Ye, Inexact preconditioned conjugate gradient method with inner-outer iterations. *SIAM J. Sci. Comput.* **21** (1999) 1305–1320.
[25] M. Grote and T. Huckle, Parallel preconditioning with sparse approximate inverses. *SIAM J. Sci. Comput.* **18** (1997) 838–853.
[26] W.Z. Gui and I. Babuška, The *h*-, *p*- and *hp*-version of the Finite Element Method in one dimension, I: The error analysis of the *p*-version, II: The error analysis of the *h*- and *hp*-version, III: The adaptive *hp*-version. *Numer. Math.* **49** (1986) 577–683.
[27] B. Guo and W. Cao, Additive Schwarz methods for the *hp* version of the finite element method in two dimensions. *SIAM J. Scientific Comput.* **18** (1997) 1267–1288.
[28] R. Henderson, Dynamic refinement algorithms for spectral element methods. *Comput. Methods Appl. Mech. Engrg.* **175** (1999) 395–411.
[29] I.C.F. Ipsen and C.D. Meyer, The idea behind Krylov methods. *Amer. Math. Monthly* **105** (1998) 889–899.
[30] G.E. Karniadakis and S. Sherwin, *Spectral/hp Element Methods for CFD*. Oxford University Press (1999).
[31] V. Korneev, J.E. Flaherty, J.T. Oden and J. Fish, Additive Schwarz algorithms for solving hp-version finite element systems on triangular meshes. *Appl. Numer. Math* **43** (2002) 399–421.
[32] V. Korneev, U. Langer and L.S. Xanthis, On fast domain decomposition solving procedures for *hp*-discretizations of 3d elliptic problems. *Comput. Methods Appl. Math.* **3** (2003) 536–559.
[33] P. Le Tallec and A. Patra, Non–overlapping domain decomposition methods for adaptive *hp* approximations of the Stokes problem with discontinuous pressure fields. *Comput. Methods Appl. Mech. Engrg.* **145** (1997) 361–379.
[34] J.W. Lottes and P.F. Fischer, *Hybrid Multigrid/Schwarz algorithms for the spectral element method.* Technical report, Mathematics and Computer Science Division, Argonne National Laboratory (January 2003).
[35] J. Mandel and M. Brezina, Balancing domain decomposition for problems with large jumps in coefficients. *Math. Comp.* **65** (1996) 1387–1401.
[36] J.M. Melenk and C. Schwab, *hp*–FEM for reaction–diffusion equations. I: Robust exponential convergence. *SIAM J. Numer. Anal.* **35** (1998) 1520–1557.
[37] M. Melenk, *hp-finite element methods for singular perturbations*. Springer Verlag. *Lect. Notes Math.* **1796** (2002).
[38] P. Monk, *Finite element methods for Maxwell's equations*. Numerical Mathematics and Scientific Computation, The Clarendon Press Oxford University Press, New York, 2003.
[39] R. Nicolaides, Deflation of conjugate gradients with application to boundary value problems. *SIAM J. Numer. Anal.* **24** (1987) 355–36.
[40] J.T. Oden, A. Patra and Y. Feng, Parallel domain decomposition solver for adaptive *hp* finite element methods. *SIAM J. Numer. Anal.* **34** (1997) 2090–2118.
[41] L.F. Pavarino, Neumann-Neumann algorithms for spectral elements in three dimensions. *RAIRO: Modél. Math. Anal. Numér.* **31** (1997) 471–493.
[42] L.F. Pavarino and O.B. Widlund, Balancing Neumann-Neumann algorithms for incompressible Navier-Stokes equations. *Commun. Pure Appl. Math.* **55** (2002) 302–335.
[43] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*. Springer-Verlag, Berlin (1994).
[44] J. Ruge and K. Stüben, Algebraic multigrid, in *Multigrid Methods*, S. Mc Cormick Ed. SIAM Philadelphia (1987) 73–130.
[45] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.* **14** (1993) 461–469.
[46] Y. Saad and M. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear system. *SIAM J. Sci. Statist. Comput.* **7** (1986) 856–869.
[47] Y. Saad and B. Suchomel, Arms: an algebraic recursive multilevel solver for general sparse linear systems. *Numer. Linear Algebra Appl.* **9** (2002) 359–378.
[48] M.V. Sarkis, *Schwarz Preconditioners for Elliptic Problems with Discontinuous Coefficients Using Conforming and Non-Conforming Elements*. Ph.D. thesis, Courant Institute, New York University, September (1994). TR671, Department of Computer Science, New York University, `URL: file://cs.nyu.edu/pub/tech-reports/tr671.ps.Z`.
[49] D. Schötzau and C. Schwab, Time discretization of parabolic problems by the *hp*-version of the discontinuous Galerkin finite element method. *SIAM J. Numer. Anal.* **38** (2000) 837–875.
[50] C. Schwab, *p– and hp– Finite Element Methods*. Oxford Science Publications (1998).
[51] C. Schwab and M. Suri, The *p* and *hp* version of the finite element method for problems with boundary layers. *Math. Comp.* **65** (1996) 1403–1429.

[52] C. Schwab, M. Suri and C.A. Xenophontos, The *hp*–FEM for problems in mechanics with boundary layers. *Comput. Methods Appl. Mech. Engrg.* **157** (1998) 311–333.
[53] B.F. Smith, P.E. Bjørstad and W.D. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press (1996).
[54] P. Solin, K. Segeth and I. Dolezel, *Higher-order finite element methods*. Studies in Advanced Mathematics, Chapman and Hall, 2004.
[55] A. Toselli, FETI *domain decomposition methods for scalar advection-diffusion problems. Comput. Methods Appl. Mech. Engrg.* **190** (2001) 5759–5776.
[56] A. Toselli and X. Vasseur, *Domain decomposition methods of Neumann-Neumann type for hp-approximations on geometrically refined boundary layer meshes in two dimensions*. Technical Report 02–15, Seminar für Angewandte Mathematik, ETH, Zürich (September 2002). Submitted to Numerische Mathematik.
[57] A. Toselli and X. Vasseur, A numerical study on Neumann-Neumann and FETI methods for *hp*-approximations on geometrically refined boundary layer meshes in two dimensions. *Comput. Methods Appl. Mech. Engrg.* **192** (2003) 4551–4579.
[58] A. Toselli and X. Vasseur, Domain decomposition methods of Neumann-Neumann type for *hp*-approximations on boundary layer meshes in three dimensions. *IMA J. Numer. Anal.* **24** (2004) 123–156.
[59] A. Toselli and O. Widlund, *Domain Decomposition methods – Algorithms and Theory*. Springer Series on Computational Mathematics, Springer **34** (2004).
[60] U. Trottenberg, C. Oosterlee and A. Schüller, *Multigrid*. Academic Press, London (2000). Guest contribution by Klaus Stüben: "An Introduction to Algebraic Multigrid".