FETI-DP preconditioners for the Virtual Element Method on general 2D meshes

Daniele Prada[1,a], Silvia Bertoluzza[1,b], Micol Pennacchio[1,c], and Marco Livesu[2]

[1] Istituto di Matematica Applicata e Tecnologie Informatiche del CNR, Via Ferrata 1, Pavia, Italy, [a]daniele.prada@imati.cnr.it, [b]silvia.bertoluzza@imati.cnr.it, [c]micol.pennacchio@imati.cnr.it

[2] Istituto di Matematica Applicata e Tecnologie Informatiche del CNR, Via dei Marini 6, Genova, Italy, marco.livesu@ge.imati.cnr.it

Abstract. We analyze the performance of a state-of-the-art domain decomposition approach, the Finite Element Tearing and Interconnecting Dual Primal (FETI-DP) method [1], for the efficient solution of very large linear systems arising from elliptic problems discretized by the Virtual Element Method (VEM) [2]. We provide numerical experiments on a model linear elliptic problem with highly heterogeneous diffusion coefficients on arbitrary Voronoi meshes, which we modify by adding nodes and edges deriving from the intersection with an unrelated coarse decomposition. The experiments confirm also in this case that the FETI-DP method is numerically scalable with respect to both the problem size and number of subdomains, and its performance is robust with respect to jumps in the diffusion coefficients and shape of the mesh elements.

1 Introduction

Polytopic meshes allow the treatment of complex geometries, a crucial task for many applications in computational engineering and scientific computing. We consider here the problem of preconditioning the Virtual Element Method (VEM), which can be viewed as an extension of the Finite Element Method to handle such a kind of meshes. In view of a possible parallel implementation of the method, we consider a state-of-the-art domain decomposition approach, the Finite Element Tearing and Interconnecting Dual Primal (FETI-DP) method. It has been proved that the FETI-DP method is still scalable when dealing with VEM, under the assumptions that the subdomains, obtained by agglomerating clusters of polygonal elements, are shape regular [3]. Such an assumption can be quite restrictive. In practice, it reduces to asking that the fine tessellation is built as a refinement of the previously existing coarse subdomain decomposition. This, of course, does not generally hold, so, in order to apply FETI-DP in a more general case, we propose to build the coarse decomposition independently from the tessellation, and modify the latter by inserting nodes and edges deriving from "cutting" it with the macro-edges of the subdomains. Of course, the resulting modified tessellation will possibly contain nasty elements with very small edges. Numerical tests do however

show that FETI-DP is quite robust in this respect, providing satisfactory results also in this framework.

This paper is organized as follows. A basic description of VEM is given in Section 2. The FETI-DP method is introduced in Section 3, whereas the algorithm for partitioning Ω into subdomains and modifying the mesh is given in Section 4. Numerical experiments that validate the theory are presented in Section 5.

2 The Virtual Element Method (VEM)

In this paper we focus on the numerical solution of the following model elliptic boundary value problem of second order discretized with VEM

$$-\nabla \cdot (\rho \nabla u) = f \text{ in } \Omega, \qquad u = 0 \text{ on } \partial\Omega, \tag{1}$$

with $f \in L^2(\Omega)$, where $\Omega \subset \mathbb{R}^2$ is a polygonal domain. We assume that the coefficient ρ is a scalar such that for almost all $x \in \Omega$, $\alpha \leq \rho(x) \leq M$ for two constants $M \geq \alpha > 0$. The variational formulation of equation (1) reads as follows: find $u \in V := H_0^1(\Omega)$ such that

$$a(u, v) = (f, v) \ \forall v \in V, \tag{2}$$

with

$$a(u, v) = \int_\Omega \rho(x) \nabla u(x) \cdot \nabla v(x)\, dx, \qquad (f, v) = \int_\Omega f(x) v(x)\, dx.$$

We consider a family $\{\mathcal{T}_h\}_h$ of tessellations of Ω into a finite number of simple polygons K, and let $\mathcal{E}_h$ be the set of edges e of $\mathcal{T}_h$. For each tessellation $\mathcal{T}_h$, we assume there exist constants $\gamma_0, \gamma_1, \alpha_0, \alpha_1 > 0$ such that:

- each element $K \in \mathcal{T}_h$ is star-shaped with respect to a ball of radius $\geq \gamma_0 h_K$, where h_K is the diameter of K;
- for each element $K \in \mathcal{T}_h$ the distance between any two vertices of K is $\geq \gamma_1 h_K$;
- $\mathcal{T}_h$ is quasi-uniform, that is, for any two elements K and K' in $\mathcal{T}_h$ we have $\alpha_0 \leq h_K / h_{K'} \leq \alpha_1$.

For each polygon $K \in \mathcal{T}_h$ we define a local finite element space $V^{K,k}$ as

$$V^{K,k} = \{v \in H^1(K) : v|_{\partial K} \in C^0(\partial K),\ v|_e \in \mathbb{P}_k(e)\ \forall e \in \mathcal{E}^K,\ \Delta v \in \mathbb{P}_{k-2}(K)\},$$

with $\mathbb{P}_{-1} = \{0\}$. Then, the global virtual element space V_h is defined as

$$V_h = \{v \in V : w|_K \in V^{K,k}\ \forall K \in \mathcal{T}_h\}.$$

We will consider the following degrees of freedom, uniquely identifying a function $v_h \in V_h$:

- the values of v_h at the vertices of the tessellation;
- for each edge e, the values of v_h at the $k-1$ internal points of the $k+1$ points Gauss-Lobatto quadrature rule on e;
- for each element K, the moments up to order $k-2$ of v_h in K.

Due to the definition of the discrete space V_h, the bilinear form a in equation (2) is not directly computable on discrete functions in terms of the degrees of freedom. The VEM stems from replacing a with a suitable approximate bilinear form a_h. Thus, the discrete form of problem 1 reads as follows: find $u_h \in V_h$ such that

$$a_h(u_h, v_h) = f_h(v_h) \qquad \forall\, v_h \in V_h. \tag{3}$$

Further details on how the bilinear form a_h, as well as the study of the convergence, stability and robustness properties of the method can be found in [4,6,5]. For further details on the implementation of the method we refer to [2].

3 The FETI-DP Domain Decomposition Method for the VEM

Since the degrees of freedom corresponding to the edges of the polygons in the tessellation are nodal values, the FETI-DP method is defined as in the finite element case. More precisely let Ω be split as $\Omega = \cup_\ell \Omega^\ell$, with $\Omega^\ell = \cup_{K \in \mathcal{T}_h^\ell} K$, where $\mathcal{T}_h^\ell$ are disjoint subsets of $\mathcal{T}_h$. In view of the quasi uniformity assumptions on the tessellation $\mathcal{T}_h$ and assuming that also the decomposition into subdomains is quasi uniform, we can introduce global mesh size parameters H and h such that for all ℓ and for all K we have $h_K \simeq h$ and $\mathrm{diam}(\Omega^\ell) \simeq H$. We let $\Gamma = \cup_\ell \partial\Omega^\ell \setminus \partial\Omega$ denote the skeleton (or interface) of the decomposition.

Let $\widetilde{V}_h \supset V_h$ denote the space obtained by dropping the continuity constraint at all nodes interior to the macro edges of the decomposition (which we will call dual nodes), while retaining continuity at cross points. Problem 3 is equivalent to finding $\tilde{u}_h \in \widetilde{V}_h$ satisfying

$$\begin{cases} J(\tilde{u}_h) = \min_{\tilde{v}_h \in \widetilde{V}_h} J(\tilde{v}_h), \text{ with } J(\tilde{v}_h) = \dfrac{1}{2} a_h(\tilde{v}_h, \tilde{v}_h) - \int_\Omega f \tilde{v}_h, \\ \text{such that } \tilde{u}_h \text{ is continuous across the interface.} \end{cases} \tag{4}$$

For each $\tilde{v}_h \in \widetilde{V}_h$, we denote by $\tilde{\boldsymbol{v}} \in \mathbb{R}^M$ the corresponding vector coefficient, where M is the dimension of $\widetilde{V}_h$. Let B be a matrix whose entries assume value in the set $\{-1, 0, 1\}$. The continuity constraints across the interface can then be expressed as $B\tilde{\boldsymbol{u}} = \boldsymbol{0}$. By introducing a set of Lagrange multipliers $\boldsymbol{\lambda} \in \mathrm{range}(B)$ to enforce the continuity constraints, we obtain a finite dimensional saddle point formulation of (4)

$$\widetilde{A}\tilde{\boldsymbol{u}} + B^T \boldsymbol{\lambda} = \tilde{\boldsymbol{f}}, \qquad B\tilde{\boldsymbol{u}} = \boldsymbol{0},$$

where $\widetilde{A}$ and $\tilde{\boldsymbol{f}}$ are the finite dimensional representation of $a_h(\tilde{v}_h, \tilde{w}_h)$ and $\int_\Omega f\tilde{v}_h$, $\forall\, \tilde{v}_h, \tilde{w}_h \in \widetilde{V}_h$, respectively. Since $\widetilde{A}$ is symmetric and positive definite [1], we can eliminate $\tilde{\boldsymbol{u}}$, and obtain a linear system for the Lagrange multiplier. This linear system is solved with a conjugate gradient method with a preconditioner that takes the same form as in the finite element case ([1]). In [3] the authors proved that, as in the finite element case, the condition number of the preconditioned matrix increases at most as $(1+\log(k^2H/h))^2$, under the assumption that the Ω^ℓ are shape regular.

4 Subdomain Partitioning by Conformal Meshing

In general, subdomains Ω^ℓ obtained as the union of polygonal elements of a tessellation are not shape regular, unless this is constructed in two stages: first, a decomposition into shape regular subdomains is defined; then, each subdomain is refined to obtain the final tessellation. An alternative is to define the subdomains independently of the tessellation and to modify the latter by "cutting" it with the edges of the subdomains. By construction, the subdomains will then be the union of elements of the new tessellation.

We provide here details of our meshing algorithm, which we implemented in C++ using CinoLib [8]. Given a tessellation $\mathcal{T}_h$ of Ω and a set of L polygonal subdomains that jointly cover Ω without overlaps, we output a domain tessellation with matching interfaces along subdomain boundaries.

In the general case the edges of $\mathcal{T}_h$ will intersect subdomain boundaries at many points. We start by splitting all the edges that are crossed by subdomain boundaries, thereby producing an edge soup. For any pair of edges in the soup, either the two edges are completely disjoint or they meet at a common endpoint (Fig. 1, middle). When computing intersections, exact arithmetics should be used to handle pathological cases due to roundoff errors introduced by finite machine arithmetics [9].

In order to generate the output, we process the edge soup and produce a list of polygons, each one represented as an ordered list of vertices. To this end, for each oriented edge in the soup, we trace polygons by iteratively visiting the leftmost edge until hitting the starting edge at its opposite endpoint, thus closing the polygon (Fig. 1, middle right). If all the edges are visited twice (both orientations count) the result is a list of polygons, with vertices ordered counterclockwise. The only polygon with clockwise winding order will be the one covering the whole domain and containing all and only the boundary edges of $\mathcal{T}_h$, which is discarded during the post-processing phase.

The leftmost edge is found as follows: given an oriented edge e (from node v_i to node v_j), the set of edges incident to it at v_j are first classified as being either on the left or on the right of e. Let $e'(v_j \to v_k)$ be one of these edges and A the 2×2 matrix having $(v_j - v_i)$ and $(v_k - v_j)$ as rows. e' is at the left of e if $\det(A) > 0$, at the right of e if $\det(A) < 0$ and collinear with e if $\det(A) = 0$. Computing $\det(A)$ is known to be a critical issue in

finite floating point arithmetics. For this reason, Shewchuck predicates [7] are used to estimate its sign robustly. If there exist some edges at the left of e, then the polygon is locally convex and the leftmost turn is the one that passes through the edge e' that minimizes the dot product between $e'/\|e'\|$ and $e/\|e\|$. Conversely, if all the edges incident to v_j are at the right of e, the polygon is locally concave and the leftmost turn is the one that passes through the edge e'' that maximizes the dot product between $e''/\|e''\|$ and $e/\|e\|$.

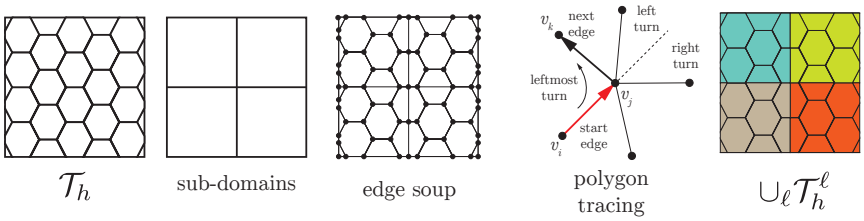

Fig. 1: Schematic representation of our meshing algorithm.

5 Numerical Results

Here we consider the performance of the preconditioned FETI-DP method. In each experiment, the domain $\Omega = [0,1]^2$ has been discretized using Voronoi cells of arbitrary shape. Every tessellation $\mathcal{T}_h$ of Ω is partitioned into L squared subdomains using the approach described in Section 4. This approach can introduce new polygons with very small edges, as shown in Fig. 2. In order to test the robustness of FETI-DP, we consider two different types of data: i) $\rho = 1, f = \sin(2\pi x)\sin(2\pi y)$; ii) for each subdomain, $\rho = 10^\alpha, \alpha$ random integer in $[-5,5]$, f uniform random in $[-1,1]$.

Table 1 shows that, with the first type of data, by fixing the initial mesh $\mathcal{T}_h$ and increasing the number of subdomains, thereby fixing h while decreasing H, the condition number κ decreases as expected. The smallest edge in the mesh $h_{\min}$, as well as the parameters γ_0 and γ_1 introduced in Section 2 are also listed to stress that the presence of arbitrarily shaped Voronoi polygons does not hinder the convergence of the method. In Fig. 3, $\kappa^{1/2}$ is plotted as a function of the degrees of freedom of the whole problem, for varying number of subdomains L. The size of the problem is increased by taking five different meshes with 10000, 20000, 40000, 80000, and 160000 polygons, respectively, as initial tessellations of Ω. Solid and dashed lines correspond to the first and second types of data, respectively. With the first type, for fixed L, $\kappa^{1/2}$ clearly grows as log(degrees of freedom), in agreement with the theoretical bound. This behavior does not seem to be affected by jumps in the diffusion coefficient, especially for high L (see Fig. 3, right). Finally, we present some

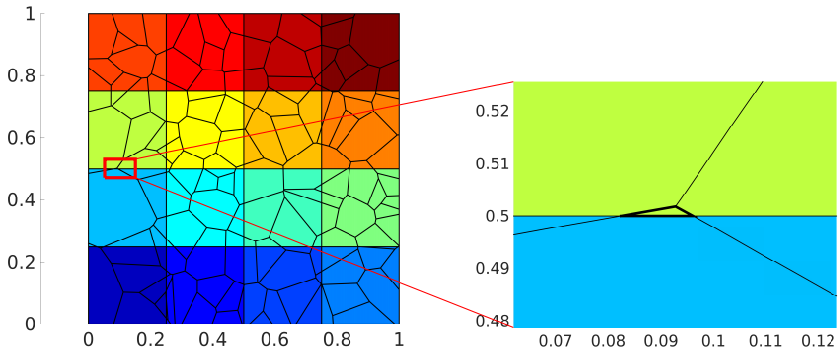

Fig. 2: (Left) General Voronoi mesh partitioned into $L = 16$ subdomains. (Right) Zoomed view of the boundary between two subdomains, where a tiny triangle is generated.

computations performed with high order elements (see Table 2 and Fig. 4). In Fig. 4 (Left), it is possible to verify the polylogarithmic dependence of κ on the polynomial order k. Indeed, for H/h fixed, the expected bound $(1+\log(k^2H/h)^2 \sim (1+log(k^2))^2 \sim (1+2\log(k))^2$ is confirmed. In Fig. 4 (Right), we keep the polynomial order fixed to $k=3$ and $k=5$ and increase the dimension of the problem by using the five meshes mentioned above. The expected bound $(1+log(k^2H/h))^2 \sim (1+log(H/h))^2$ (for fixed k) is confirmed for both types of problem data.

All these experiments demonstrate that the performance of the preconditioned FETI-DP method is robust with respect to jumps in the diffusion coefficients and shape of the mesh elements.

Table 1: Results obtained with the first type of data and polynomial order $k=1$ on two Voronoi meshes.

voro1, 40000 initial polygons								
L	D.o.f.	$1/h$	$h_{\min}$	γ_0	γ_1	$\lambda_{\min}$	$\lambda_{\max}$	It.
64	86 202	55.93	1.04×10^{-5}	5.73×10^{-3}	1.32×10^{-3}	1.05	5.78	14
144	90 561	55.93	1.04×10^{-5}	3.97×10^{-3}	1.32×10^{-3}	1.05	5.37	14
256	94 954	55.93	1.04×10^{-5}	2.94×10^{-3}	1.13×10^{-3}	1.06	4.90	13
voro2, 160000 initial polygons								
L	D.o.f.	$1/h$	$h_{\min}$	γ_0	γ_1	$\lambda_{\min}$	$\lambda_{\max}$	It.
64	330 104	113.38	1.01×10^{-5}	4.11×10^{-4}	2.47×10^{-3}	1.06	7.32	16
144	338 451	113.38	1.04×10^{-5}	4.11×10^{-4}	2.10×10^{-3}	1.05	6.56	15
256	346 829	113.38	1.00×10^{-5}	4.11×10^{-4}	1.87×10^{-3}	1.05	6.25	15

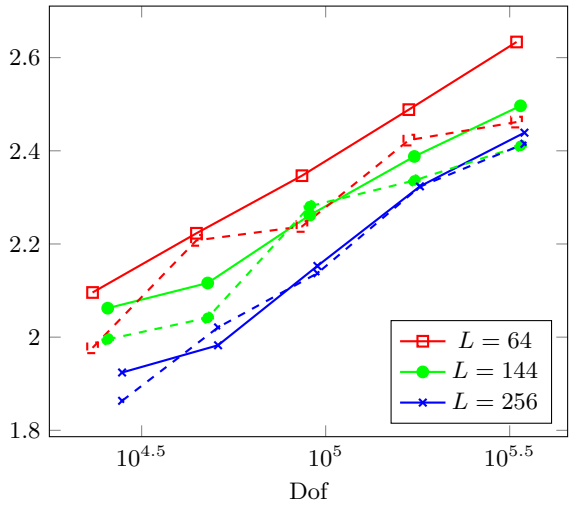

Fig. 3: Plots of $\kappa^{1/2}$ as a function of the global degrees of freedom for fixed polynomial order $k = 1$ but increasing number of subdomains L. Solid and dashed lines correspond to the first and second types of data, respectively.

Table 2: $\kappa^{1/2}$ and number of iterations of the preconditioned FETI-DP system with the first type of data, for a fixed starting mesh (voro2, see Table 1) but increasing number of subdomains L and polynomial order k.

$L\backslash k$	2		3		4		5		6	
64	3.13	19	3.51	21	3.75	23	3.91	23	4.05	24
144	2.98	18	3.37	21	3.60	22	3.78	23	3.93	24
256	2.97	18	3.37	21	3.57	22	3.74	23	3.88	24

Acknowledgments

This paper has been realized in the framework of the ERC Project CHANGE, which has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No 694515). The authors would also like to thank the members of the Shapes and Semantics Modeling Group at IMATI-CNR for fruitful discussions on conformal meshing.

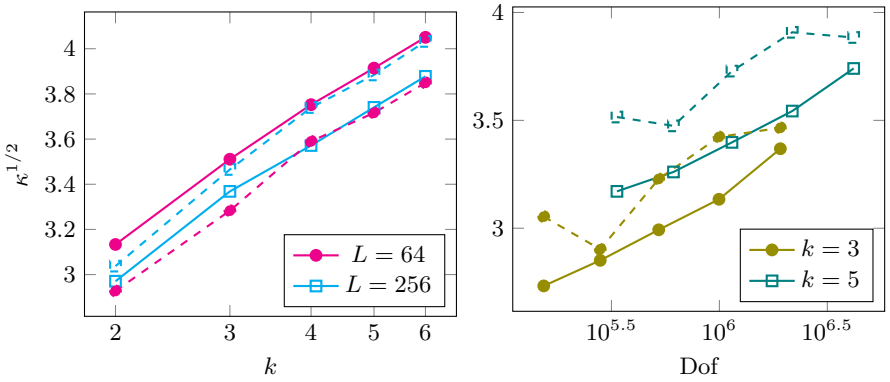

Fig. 4: High order elements. Solid and dashed lines correspond to the first and second types of problem data, respectively. (Left) Plot of $\kappa^{1/2}$ as a function of the polynomial order k, initial mesh voro2 (see Table 1). We have $H/h \approx 20.04$ for $L = 64$ and $H/h \approx 10.02$ for $L = 256$. (Right) Plot of $\kappa^{1/2}$ as a function of the global degrees of freedom, $H/h \approx 10.02, L = 256$.

References

1. A. Toselli and O. Widlund, *Domain Decomposition Methods - Algorithms and Theory*. Springer Series in Computational Mathematics **34** (2005).
2. L. Beirão da Veiga, F. Brezzi, L. D. Marini and A. Russo, *The hitchhiker's guide to the Virtual Element Method*, Math Models Methods Appl Sci **24** (2014).
3. S. Bertoluzza, M. Pennacchio and D. Prada, *BDDC and FETI-DP for the virtual element method*, Calcolo **54**:4 (2017), 1565–1593.
4. L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini and A. Russo *Basic principles of virtual element methods*, Math Models Methods Appl Sci **23**:1 (2013), 199–214.
5. L. Beirão da Veiga, A. Chernov, L. Mascotto and A. Russo, *Basic principles of hp virtual elements on quasiuniform meshes*, Math Models Methods Appl Sci **26**:8 (2016), 1567–1598.
6. L. Beirão da Veiga, F. Brezzi, L. D. Marini and A. Russo, *Virtual Element Method for general second-order elliptic problems on polygonal meshes*, Math Models Methods Appl Sci **26**:4 (2016), 729–750.
7. J. R. Shewchuk, *Adaptive precision floating-point arithmetic and fast robust geometric predicates*, Discrete Comput Geom **18**:3 (1997), 305–363.
8. M. Livesu, *CinoLib: a generic programming header only C++ library for processing polygonal and polyhedral meshes*, https://github.com/maxicino/cinolib.
9. M. Attene, *Direct repair of self-intersecting meshes*, Graph Models **76**:6 (2014), 658–668.