

The Reduced Basis Multigrid Scheme for the Virtual Element Method

Paola F. Antonietti^[0000-0002-2138-3878],
Silvia Bertoluzza^[0000-0002-1354-7158],
Fabio Credali^[0000-0001-8568-7021]

Introduction

The virtual element method (VEM) was introduced more than a decade ago as a generalization of the finite element method on general polytopal meshes [7]. One of the main features of VEM is that the underlying discrete space is defined implicitly. Indeed, the basis functions are themselves solutions of a local differential problem in each mesh element. Such a local partial differential equation (PDE) is generally not solved explicitly, and the VEM discrete problem is defined using suitable polynomial projectors and stabilization terms. However, this construction presents some drawbacks: for instance, a wrong choice of the stabilization term may pollute the results [4]. From the algebraic solver perspective, the design of geometric multigrid schemes is non-trivial since the Virtual Element spaces are non-nested even if built on a sequence of nested meshes, and the injection operator cannot be defined by simple pointwise evaluation of basis functions at points located in the interior of the elements [1]. To overcome the issues mentioned above, methods for reconstructing the virtual basis functions [5, 6, 3] have been introduced in the literature.

This work aims to exploit the reduced basis Virtual Element Method (rbVEM) we introduced in [5] to construct a geometric W-cycle multigrid scheme for the lowest order virtual element method on non-nested meshes. To the best of our knowledge,

Paola F. Antonietti
MOX, Dipartimento di Matematica, Politecnico di Milano, 20133 Milano, Italy, e-mail: paola.antonietti@polimi.it

Silvia Bertoluzza
IMATI “E. Magenes”, 27100 Pavia, Italy, e-mail: silvia.bertoluzza@imati.cnr.it

Fabio Credali
King Abdullah University of Science and Technology, 23955 Thuwal, Saudi Arabia, e-mail: fabio.credali@kaust.edu.sa

this is the first work to exploit the efficient computation of the virtual basis function to design a multigrid algorithm for the h -version of VEM.

We consider the standard virtual element formulation for the Poisson problem and the rbVEM space as an auxiliary tool for constructing the intergrid operators between mesh levels. In particular, we define the prolongation operator as the L^2 -projection [2] between rbVEM spaces on different levels, without the need of modifying the original discrete problem. We present numerical tests showing that the convergence factor of our new scheme is independent of the resolution of the underlying fine mesh. We also show that our algorithm is independent of the number of reduced basis functions employed for constructing the auxiliary rbVEM space. A forthcoming work will address a detailed convergence analysis for the proposed method, as well as the design of V-cycle schemes.

1 Model problem and Virtual Element discretization

We consider the weak formulation of the Poisson equation with homogeneous Dirichlet boundary conditions in a two-dimensional polygonal domain $\Omega \subset \mathbb{R}^2$, which reads: given $f \in L^2(\Omega)$, find $u \in H_0^1(\Omega)$ such that

$$\mathcal{A}(u, v) = (\nabla u, \nabla v)_\Omega = (f, v)_\Omega \quad \forall v \in H_0^1(\Omega), \quad (1)$$

where the notation $(\cdot, \cdot)_\Omega$ indicates the scalar product in $L^2(\Omega)$.

We discretize our problem by considering the virtual element method. Let \mathcal{T}_h be a generic polygonal tessellation of Ω made of disjoint star-shaped polygonal elements K with diameter h_K . The size of \mathcal{T}_h is denoted by $h = \max_{K \in \mathcal{T}_h} h_K$. The local virtual element space [7] of degree one is defined as $V(K) = \{v \in H^1(K) : v|_\varepsilon \in \mathbb{P}_1(\varepsilon) \forall \varepsilon \in \partial K, \Delta v = 0 \text{ in } K\}$, and the global discrete space is then obtained by gluing all local spaces by continuity

$$V(\mathcal{T}_h) = \{v \in H_0^1(\Omega) : v \in V(K) \forall K \in \mathcal{T}_h\}.$$

We observe that a function $v \in V(K)$ is uniquely identified by its values at the N vertices of K , which are a unisolvent set of degrees of freedom. Moreover, accuracy is guaranteed since $\mathbb{P}_1(K) \subset V(K)$.

In order to solve Equation (1) by means of a geometric multigrid algorithm, we consider a sequence of quasi-uniform tessellations $\{\mathcal{T}_j\}_{j=1}^J$ with mesh sizes h_j satisfying the constraint $C h_{j-1} \leq h_j \leq h_{j-1}$, for a positive constant C . Moreover, $\{\mathcal{T}_j\}_{j=1}^J$ are not-nested. We do not assume that meshes are nested, as the nestedness of the meshes does not imply that of the corresponding spaces. We then seek for the virtual element approximation of u at the finest level grid J :

$$\text{find } u_J \in V(\mathcal{T}_J) \text{ such that } \mathcal{A}_J(u_J, v_J) = (f_J, v_J)_\Omega \quad \forall v_J \in V(\mathcal{T}_J). \quad (2)$$

The discrete bilinear form \mathcal{A}_J is defined as

$$\begin{aligned}\mathcal{A}_J(u_J, v_J) &= \sum_{K_J \in \mathcal{T}_J} \mathcal{A}_J^{K_J}(u_J, v_J), \\ \mathcal{A}_J^{K_J}(u_J, v_J) &= (\nabla \Pi^\nabla u_J, \nabla \Pi^\nabla v_J)_{K_J} + S_J^{K_J}((\text{Id} - \Pi^\nabla)u_J, (\text{Id} - \Pi^\nabla)v_J),\end{aligned}$$

where Π^∇ is the elliptic projection operator onto $\mathbb{P}_1(K_J)$ solving

$$(\nabla v_J - \nabla \Pi^\nabla v_J, \nabla q)_{K_J} = 0 \quad \forall q \in \mathbb{P}_1(K_J), \quad \int_{\partial K_J} \Pi^\nabla v_J ds = \int_{\partial K_J} v_J ds,$$

and $S_J^{K_J}$ is the semi-positive definite bilinear form defined as

$$S_J^{K_J}(w_J, v_J) = \sum_{i=1}^N w_J(\mathbf{x}_i) v_J(\mathbf{x}_i), \quad (3)$$

with $\mathbf{x}_1, \dots, \mathbf{x}_N$ being the vertices of $K_J \in \mathcal{T}_J$. Finally, the discrete right-hand side is defined as $(f_J, v_J)_\Omega = \sum_{K_J \in \mathcal{T}_J} (\Pi_0^0 f, v_J)_{K_J}$, where Π_0^0 is the L^2 projection onto constants, i.e., $\Pi_0^0 v_J = |K_J|^{-1} \int_{K_J} v_J dx$ for any $v_J \in V(K_J)$. Eq. (2) can be reformulated in terms of operators as $A_J u_J = f_J$.

2 The reduced basis multigrid algorithm

In this section we design a W-cycle non-nested multigrid algorithm to solve Eq. (2). A key ingredient for the construction of the method are the intergrid operators. Since $V(\mathcal{T}_{j-1}) \not\subset V(\mathcal{T}_j)$, a good choice of prolongation operator is the L^2 projection [2] $I_{j-1}^j : V(\mathcal{T}_{j-1}) \rightarrow V(\mathcal{T}_j)$ defined, for $v_{j-1} \in V(\mathcal{T}_{j-1})$, as

$$(I_{j-1}^j v_{j-1}, w_j)_\Omega = (v_{j-1}, w_j)_\Omega \quad \forall w_j \in V(\mathcal{T}_j).$$

Clearly, due to the implicit definition of the virtual element spaces, the quantities above are not fully computable through the degrees of freedom. We thus consider the rbVEM space introduced in [5], whose basis functions are explicitly constructed and have the same degrees of freedom as the original VEM space. For a generic element $K_j \in \mathcal{T}_j$, the rbVEM is defined as

$$V^{\text{rb}}(K_j) = \mathbb{P}_1(K_j) \oplus W^{\text{rb}}(K_j), \quad (4)$$

where $W^{\text{rb}}(K_j) \subset \text{span}\{e_{M,i}^{\text{rb}}, i = 1, \dots, N\}$ is such that $w_j^{\text{rb}} = \sum_{i=1}^N \mathbf{w}_i e_{M,i}^{\text{rb}} \in W^{\text{rb}}(K_j)$ if and only if $\Pi^\nabla(\sum_{i=1}^N \mathbf{w}_i e_i) = 0$, with $\{e_i\}_{i=1}^N$ being the shape functions of $V(K_j)$. The functions $e_{M,i}^{\text{rb}}$ are an approximation of the virtual element basis functions computed by means of the reduced basis method with complexity, i.e. the dimension of the reduced basis space, equal to M . By construction, $e_{M,i}^{\text{rb}}(\mathbf{x}_k) = \delta_{i,k}$ for $i, k = 1, \dots, N$, see [5]. In addition, the space $W^{\text{rb}}(K_j)$ is constructed to be

\mathcal{A} -orthogonal to $\mathbb{P}_1(K_j)$. Given $v_j \in V(K_j)$, its counterpart $v_j^{\text{rb}} \in V^{\text{rb}}(K_j)$ is constructed as $v_j^{\text{rb}} = \Pi^\nabla v_j + v_j^\perp$, where $v_j^\perp \in W^{\text{rb}}(K_j)$ is the linear combination $v_j^\perp = \sum_{i=1}^N (v_j(\mathbf{x}_i) - \Pi^\nabla v_j(\mathbf{x}_i)) e_{M,i}^{\text{rb}}$. Given $v_j, w_j \in V(K_j)$ and their counterparts $w_j^{\text{rb}}, v_j^{\text{rb}} \in V^{\text{rb}}(K_j)$, we set

$$\begin{aligned} \mathcal{A}_j^{K_j}(w_j^{\text{rb}}, v_j^{\text{rb}}) &= (\nabla \Pi^\nabla w_j, \nabla \Pi^\nabla v_j)_{K_j} + S_j^{K_j}(w_j^\perp, v_j^\perp) \\ &= (\nabla \Pi^\nabla w_j, \nabla \Pi^\nabla v_j)_{K_j} + \sum_{i=1}^N w_j^\perp(\mathbf{x}_i) v_j^\perp(\mathbf{x}_i) \\ &= (\nabla \Pi^\nabla w_j, \nabla \Pi^\nabla v_j)_{K_j} + \sum_{i=1}^N [(w_j - \Pi^\nabla w_j)(\mathbf{x}_i)] [(v_j - \Pi^\nabla v_j)(\mathbf{x}_i)] \\ &= (\nabla \Pi^\nabla w_j, \nabla \Pi^\nabla v_j)_{K_j} + S_j^{K_j}((\text{Id} - \Pi^\nabla)w_j, (\text{Id} - \Pi^\nabla)v_j) = \mathcal{A}_j^{K_j}(w_j, v_j), \end{aligned}$$

where we exploited that, for $k = 1, \dots, N$,

$$\begin{aligned} v_j^\perp(\mathbf{x}_k) &= \sum_{i=1}^N [v_j(\mathbf{x}_i) - \Pi^\nabla v_j(\mathbf{x}_i)] e_{M,i}^{\text{rb}}(\mathbf{x}_k) = \sum_{i=1}^N [v_j(\mathbf{x}_i) - \Pi^\nabla v_j(\mathbf{x}_i)] \delta_{i,k} \\ &= v_j(\mathbf{x}_k) - \Pi^\nabla v_j(\mathbf{x}_k) \end{aligned}$$

and, analogously, $w_j^\perp(\mathbf{x}_k) = w_j(\mathbf{x}_k) - \Pi^\nabla w_j(\mathbf{x}_k)$. As for the standard VEM space, the global rbVEM space is obtained by gluing all local spaces by continuity, that is

$$V^{\text{rb}}(\mathcal{T}_j) = \{v_j^{\text{rb}} \in H_0^1(\Omega) : v_j^{\text{rb}} \in V^{\text{rb}}(K_j) \forall K_j \in \mathcal{T}_j\}, \quad j = 1, \dots, J.$$

We then solve Eq. (2) in the original VEM space, by employing the newly introduced rbVEM as a support for computing intergrid operators. Given $v_{j-1}^{\text{rb}} \in V^{\text{rb}}(\mathcal{T}_{j-1})$, we define the prolongation operator $I_{j-1}^j : V^{\text{rb}}(\mathcal{T}_{j-1}) \rightarrow V^{\text{rb}}(\mathcal{T}_j)$ as

$$(I_{j-1}^j v_{j-1}^{\text{rb}}, w_j^{\text{rb}})_\Omega = (v_{j-1}^{\text{rb}}, w_j^{\text{rb}})_\Omega \quad \forall w_j^{\text{rb}} \in V^{\text{rb}}(\mathcal{T}_j), \quad (5)$$

while the restriction operator $I_j^{j-1} : V^{\text{rb}}(\mathcal{T}_j) \rightarrow V^{\text{rb}}(\mathcal{T}_{j-1})$ is defined as its adjoint

$$(I_j^{j-1} w_j^{\text{rb}}, v_{j-1}^{\text{rb}})_\Omega = (w_j^{\text{rb}}, v_{j-1}^{\text{rb}})_\Omega \quad \forall v_{j-1}^{\text{rb}} \in V^{\text{rb}}(\mathcal{T}_{j-1}). \quad (6)$$

In our scheme, the prolongation and restriction operators between the VEM spaces $V(\mathcal{T}_{j-1})$, $V(\mathcal{T}_j)$ inherit their definition from (5) and (6), respectively. We point out that the scalar product on the left-hand side of (5), which involves functions in $V^{\text{rb}}(\mathcal{T}_j)$, can be efficiently computed by exploiting the properties of the reduced basis method [5]. On the other hand, the scalar product at the right-hand side involves functions defined on two different mesh levels and requires the computation of the intersection between \mathcal{T}_j and \mathcal{T}_{j-1} , as commented in [2]. Two meshes can be efficiently intersected if a bounding box technique is employed to detect cell-

Algorithm 1 Multigrid W-cycle iteration

```

Initialize  $u_0 \in V(\mathcal{T}_J)$ :
for  $k = 0, 1, \dots$  do
     $u_{k+1} = \text{MG}_{\mathcal{W}}(J, f_J, u_k, m_1, m_2)$ ;
     $u_k = u_{k+1}$ ;
end for

```

by-cell interactions. As a smoothing scheme, we choose the Richardson iteration. Given $V^*(\mathcal{T}_j)$ the dual space of $V(\mathcal{T}_j)$, let $A_j : V(\mathcal{T}_j) \rightarrow V^*(\mathcal{T}_j)$ be the operator defined as $(A_j w, v)_\Omega = \mathcal{A}_j(w, v)$ for all $w, v \in V(\mathcal{T}_j)$ and let $\Lambda_j \in \mathbb{R}$ denote the maximum eigenvalue of A_j , for $j = 2, \dots, J$. The smoothing scheme is then defined as $B_j = \Lambda_j \text{Id}_j$, $j = 2, \dots, J$. The W-cycle iteration for computing u_j is summarized

Algorithm 2 Multigrid W-cycle scheme: one iteration at level $j \geq 1$

```

if  $j = 1$  then
     $\text{MG}_{\mathcal{W}}(1, g, z_0, m_1, m_2) = A_1^{-1}g$ .
else
    for  $i = 1, \dots, m_1$  do (Pre-smoothing)
         $z^{(i)} = z^{(i-1)} + B_j^{-1}(g - A_j z^{(i-1)})$ ;
    end for

     $r_{j-1} = I_j^{j-1}(g - A_j z^{(m_1)})$ ; (Coarse grid correction)
     $\bar{e}_{j-1} = \text{MG}_{\mathcal{W}}(j-1, r_{j-1}, 0, m_1, m_2)$ ;
     $e_{j-1} = \text{MG}_{\mathcal{W}}(j-1, r_{j-1}, \bar{e}_{j-1}, m_1, m_2)$ ;
     $z^{(m_1+1)} = z^{(m_1)} + I_{j-1}^j e_{j-1}$ ;

    for  $i = m_1 + 2, \dots, m_1 + m_2 + 1$  do (Post-smoothing)
         $z^{(i)} = z^{(i-1)} + B_j^{-1}(g - A_j z^{(i-1)})$ ;
    end for

     $\text{MG}_{\mathcal{W}}(j, g, z_0, m_1, m_2) = z^{(m_1+m_2+1)}$ .
end if

```

in Algorithm 1, where $u_0 \in V(\mathcal{T}_J)$ denotes a suitable initial guess, and where $m_1, m_2 \in \mathbb{N}$ denote the number of pre- and post-smoothing steps, respectively. We observe that $\text{MG}_{\mathcal{W}}(J, f_J, u_k, m_1, m_2)$ represents the approximate solution obtained after one iteration of the non-nested W-cycle, which is defined by the recursive procedure described in Algorithm 2. Indeed, the W-cycle is defined by induction: given the general problem of finding $z \in V(\mathcal{T}_j)$ satisfying

$$A_j z = g \tag{7}$$

for $j \geq 2$ and $g \in L^2(\Omega)$, then $\text{MG}_{\mathcal{W}}(j, g, z_0, m_1, m_2)$ is the approximate solution of (7) obtained after one multigrid iteration with initial guess $z_0 \in V(\mathcal{T}_j)$ and m_1, m_2 smoothing steps. Clearly, on the coarsest level $j = 1$, the problem is solved exactly.

Remark 1 In this work, we are using the spaces $V^{\text{rb}}(\mathcal{T}_j)$, $j = 1, \dots, J$ as an auxiliary tool for computing intergrid operators. At the same time, the discrete problem is for-

mulated in the standard virtual element space $V(\mathcal{T}_J)$. We observe that, if the reduced basis stabilization introduced in [5] is considered, then Eq. (2) can be reformulated in $V^{\text{rb}}(\mathcal{T}_J)$, so that $\mathcal{A}_J(w_J^{\text{rb}}, v_J^{\text{rb}}) = \mathcal{A}(w_J^{\text{rb}}, v_J^{\text{rb}})$ for $w_J^{\text{rb}}, v_J^{\text{rb}} \in V^{\text{rb}}(\mathcal{T}_J)$. We thus obtain a discrete formulation in the fully-conforming rbVEM space, and our multigrid scheme remains unchanged. As discussed in [5], the use of rbVEM as a discretization method would require more computational effort than computing the stabilization term in (3), which is extremely cheap. However, this overhead is not worthwhile for isotropic problems, as VEM and rbVEM exhibit the same approximation properties. Conversely, rbVEM is more accurate than standard VEM when solving anisotropic problems, which are out of the scope of the present work. We finally note that the proposed Reduced Basis Multigrid scheme is non-intrusive, as it does not require any modifications to existing VEM and Multigrid implementations.

3 Numerical results

In this section, we present numerical tests to assess the performance of the proposed multigrid scheme. We set $\Omega = (0, 1)^2$, and we consider the sets of non-nested meshes shown in Figure 1. The finest meshes (Level 4) are made of 512 (Set 1), 1024 (Set 2), 2048 (Set 3), and 4096 (Set 4) elements. We remark that each mesh was generated independently of the others. Moreover, the ratio between the numbers of elements of a coarse mesh and of the corresponding fine mesh is equal to 1/4. In the following, we report the iteration counts needed to reduce the relative residual below the tolerance $tol = 10^{-8}$. We also compute the convergence factor

$$\rho_J = \exp\left(\frac{1}{n_{it,J}} \log \frac{\|\mathbf{r}_{n_{it,J}}\|}{\|\mathbf{r}_0\|}\right),$$

where $n_{it,J}$ is the iteration count for reducing the error below the considered tolerance by means of the W-cycle scheme with J levels ($J = 2, 3, 4$), while $\mathbf{r}_{n_{it,J}}$ and \mathbf{r}_0 are the final and initial residual vectors, respectively. In all our tests, we set $m_1 = m_2 = m$. We first assess the independence of the multigrid scheme from the number M of reduced basis functions employed to construct the shape functions of the space $W^{\text{rb}}(K_j)$ in (4). By looking at Table 1, it is clear that the value of M does not affect the convergence of the proposed scheme: indeed, for fixed m and for a fixed number of levels J , the iteration count is the same for $M = 1$ and $M = 50$. For the next test, we thus set $M = 1$. The second test studies the convergence properties of our W-cycle scheme. In Table 2, we report both the convergence factor and the iteration count for all sets of meshes by varying the number m of pre- and post-smoothing steps. It is clear that the convergence factor does not depend on the mesh size h . Indeed, for fixed $J = 2, 3, 4$ and $m = 3, 6, 8$, the convergence factor is roughly constant. This fact implies that the iteration counts needed by the W-cycle to reduce the residual below the given tolerance are independent of the resolution of the underlying mesh. Moreover, as m increases, the convergence factor decreases.

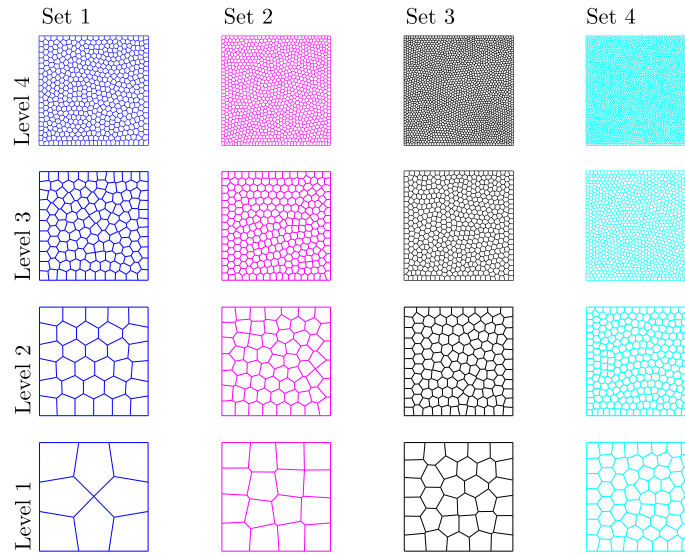


Fig. 1 Sets of non-nested grids employed for numerical tests.

Table 1 Iterations count of the W-Cycle multigrid as a function of m . Comparison in terms of the number of reduced basis functions used to build the auxiliary space: $M = 1$ vs $M = 50$.

W-cycle, Richardson smoother, $tol = 10^{-8}$												
M	Set 1						Set 2					
	2 levels		3 levels		4 levels		2 levels		3 levels		4 levels	
	1	50	1	50	1	50	1	50	1	50	1	50
$m = 3$	8	8	9	9	8	8	8	8	8	8	8	8
$m = 6$	7	7	7	7	6	6	6	6	6	6	6	6
$m = 8$	6	6	5	5	5	5	6	6	5	5	5	5

M	Set 3						Set 4					
	2 levels		3 levels		4 levels		2 levels		3 levels		4 levels	
	1	50	1	50	1	50	1	50	1	50	1	50
$m = 3$	8	8	8	8	8	8	8	8	8	8	8	8
$m = 6$	6	6	6	6	6	6	6	6	6	6	6	6
$m = 8$	5	5	5	5	5	5	5	5	5	5	5	5

4 Conclusions

We introduced a non-nested W-cycle multigrid scheme for the lowest order virtual element method based on the rbVEM auxiliary space introduced in [5]. Numerical tests showed that the W-cycle multigrid scheme is scalable and that its convergence factor does not depend on the number of reduced basis functions used to construct the auxiliary space. A forthcoming paper will present an exhaustive convergence

Table 2 Convergence factor ρ and iterations count of the W-Cycle multigrid as a function of m . The auxiliary space is constructed using a single reduced basis function.

W-cycle, Richardson smoother, $tol = 10^{-8}$, $M = 1$						
	Set 1			Set 2		
	2 levels	3 levels	4 levels	2 levels	3 levels	4 levels
$m = 3$	0.0958 (8)	0.1174 (9)	0.0964 (8)	0.0843 (8)	0.0896 (8)	0.0991 (8)
$m = 6$	0.0564 (7)	0.0515 (7)	0.0408 (6)	0.0428 (6)	0.0364 (6)	0.0392 (6)
$m = 8$	0.0306 (6)	0.0246 (5)	0.0217 (5)	0.0310 (6)	0.0204 (5)	0.0199 (5)
	Set 3			Set 4		
	2 levels	3 levels	4 levels	2 levels	3 levels	4 levels
$m = 3$	0.0805 (8)	0.0825 (8)	0.0942 (8)	0.0977 (8)	0.0796 (8)	0.0984 (8)
$m = 6$	0.0369 (6)	0.0308 (6)	0.0384 (6)	0.0418 (6)	0.0323 (6)	0.0397 (6)
$m = 8$	0.0229 (5)	0.0174 (5)	0.0196 (5)	0.0211 (5)	0.0128 (5)	0.0188 (5)

analysis of the proposed method, as well as extensions to the V-cycle algorithm and the application of smoothing schemes other than the Richardson iteration.

Acknowledgements This work was partially funded by MUR–PRIN/PNRR Bando 2022 (grant P2022BH5CB). PFA is partially supported by the European Union (ERC SyG, NEMESIS, project number 101115663). Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. The authors are members of INdAM–GNCS.

References

1. Antonietti, P.F., Berrone, S., Busetto, M., Verani, M.: Agglomeration-based geometric multigrid schemes for the virtual element method. *SIAM J. Numer. Anal.* **61**(1), 223–249 (2023)
2. Antonietti, P.F., Pennesi, G.: V-cycle multigrid algorithms for discontinuous Galerkin methods on non-nested polytopic meshes. *J. Sci. Comput.* **78**(1), 625–652 (2019)
3. Berrone, S., Pintore, M., Teora, G.: The lowest-order neural approximated virtual element method on polygonal elements. *Comput. Struct.* **314**, 107753 (2025)
4. Boffi, D., Gardini, F., Gastaldi, L.: Approximation of PDE eigenvalue problems involving parameter dependent matrices. *Calcolo* **57**(4), 41 (2020)
5. Credali, F., Bertoluzza, S., Prada, D.: Reduced basis stabilization and post-processing for the virtual element method. *Comput. Methods Appl. Mech. Eng.* **420**, 116693 (2024)
6. Trezzi, M., Zerbinati, U.: When rational functions meet virtual elements: the lightning virtual element method. *Calcolo* **61**(3), 35 (2024)
7. Beirão da Veiga, L., Brezzi, F., Cangiani, A., Manzini, G., Marini, L.D., Russo, A.: Basic principles of virtual element methods. *Math. Models Methods Appl. Sci.* **23**(01), 199–214 (2013)