Principles of a network element method

Julien Coatléven

Principles of a network element method [1]

Julien Coatléven[a]

[a]*IFP Énergies nouvelles, 1 et 4 avenue de Bois-Préau, 92852 Rueil-Malmaison, France*

Abstract

We introduce a new variational numerical method which does not require any background mesh to compute the scheme coefficients. Replacing the mesh by a point cloud endowed with connectivity, which we call a discretization network, and following the virtual element framework we derive a consistent, coercive and stable numerical scheme. We illustrate the good behavior of the method on several numerical examples.

Keywords: Poisson equation, meshless methods, variational methods, virtual element method, mimetic finite differences
2010 MSC: 65N30, 65N12, 65N15

Introduction

The mimetic technology [1] has brought to the community extremely simple, efficient and elegant ways to handle probably all classical partial differential equations, with general coefficients, even on very distorted or exotic meshes. One of its most recent versions, the virtual element method (VEM, [2]) is in fact so well designed that it even leads to much simpler proofs than its closest ancestor, the mimetic finite differences, letting us go back to the familiar variational setting of the finite element framework. The mimetic technology has also given birth to useful alternative methods such as the Hybrid-High-Order (HHO, [3, 4]) or the original VAG scheme [5] (see [6] for its formal equivalence with a VEM approach), sharing consequently the same ability to cope with generic meshes. Starting from the usual Poisson problem, all these methods have been extended to linear [7, 8] and non linear [9, 10] elasticity, parabolic problems [11], multiphase flow problems [12], Stokes problem [13], etc. The literature based on those methods is in fact already tremendous, reflecting the strong interest and enthusiasm of the mathematical community.

Those methods can handle meshes so general, it is so complex to design a mesh that will make them fail, that we can wonder if we still really need a mesh, or more precisely, what are the quantities usually provided by a mesh that are really needed to perform an accurate numerical simulation. This is the question that has led to the present work. Looking at the virtual element and HHO methods from a purely algebraic point of view, we see that the key of their success is the clear separation between consistency and stability (a property which they share with other approaches for handling general meshes, like discontinuous Galerkin methods, see [14]). While consistency has to retain certain properties of the original problem and its geometry, coercivity and thus stability can be obtained almost automatically using the virtual element principles [15]. Thus, it is natural to focus on the geometric elements that are absolutely necessary to ensure the consistency of a method. This approach naturally led us to the notion of discretization network, which we can sum up as a mesh reduced to its most basic expression: a point cloud endowed with a connectivity. This network and some weights computed on it that will correspond to a network geometry, allows to build a consistent approximation of our model Poisson problem. Then, applying virtual element principles, stability of the discrete variational formulation will indeed prove to be easy to obtain.

Discretization networks are common objects underlying many meshless methods, even if they sometimes remain implicit as in the case of partition of unity finite element methods (see [16]). The network geometry is a family of weights satisfying conditions that allows to approximate differential operators and reproduce polynomials on point clouds, and

[1]To cite this article please use: Principles of a network element method, J. Coatléven, Journal of Computational Physics, Vol. 433, 2021, https://doi.org/10.1016/j.jcp.2021.110197

are common objects for meshless methods not based on analytic partitions of unity (see [17, 18, 19, 20, 21, 22]). Generating a network geometry basically consists in generating weights allowing polynomial consistency using the chosen degrees of freedom, up to a given order. However, if this is enough to obtain the consistency of a meshfree method, as explained for instance in [20] this is not enough to obtain a compatible discretization like the one provided by the VEM on meshes. The crucial missing ingredient is called the compatibility of the discretization scheme, or in other words that Stokes' formula must be satisfied in some sense at the discrete level. This is in fact the main difficulty of generating network geometries. To circumvent it, for both technical and practical reasons in the present work we have chosen to use a slightly more general notion of geometry than those based on exact polynomial consistency and conservation property, by allowing those properties to be satisfied only approximately. Finally, contrary to the aforementioned methods of [17, 18, 19, 20, 21, 22] which are roughly speaking finite volume methods, the scheme presented here is based on the VEM approach and is thus variational by construction and, at least to the author's knowledge, it is completely original (we will call it the network element method). As a consequence, although it is very natural to define the degrees of freedom of the scheme at the points defining the discretization network as in [17] or [22], here as we are trying to mimic the VEM it will in fact be more natural to follow the lines of [18, 19, 20, 21] and consider degrees of freedom located at the interfaces between points using the connectivity of the network. Compared to other meshless approaches and in particular those based on analytic partitions of unity, it is delicate to claim that the method is completely meshless. Indeed if no mesh is used, not even to compute quadratures, we replace it by a discretization network and an associated geometry, whose practical computation is not as cheap as simply generating a point cloud. We refer the reader to [23, 24, 25] for a review of the huge literature on classical meshless methods.

The paper will be organized as follows: in the first part of the paper (section 1), we will give a general definition of discretization network and introduce our slightly generalized notion of network geometry. Then, in the second part of the paper (section 2) we will explain how to approximate a model Poisson problem using the network element method and study the consistency and stability properties of the corresponding scheme. Finally, in the last part of the paper (section 3) we exhibit some numerical results to illustrate the good behavior of the method.

1. Discretization networks and network geometry

Let Ω be an open bounded connected subset of $\mathbb{R}^d$, $d \in \mathbb{N}\backslash\{0\}$, assumed to be at least Lipschitz. For any $\boldsymbol{x} \in \mathbb{R}^d$ and any $r > 0$, we denote $B(\boldsymbol{x}, r)$ the ball of radius r centered at $\boldsymbol{x}$. We now define a discretization network as a point cloud endowed with a connectivity (see figure 1 for an example of discretization network), which is basically a reformulation of the networks underlying the methods of [18, 19, 20]:

Definition 1.1 (Discretization network). A discretization network $\mathcal{N}$ of Ω is defined from two sets of points $\mathcal{P}_\mathcal{T}$ and $\mathcal{P}_\mathcal{F}$, by setting $\mathcal{N} = \{\mathcal{T}, \mathcal{F}\}$, where:

- The set of cells $\mathcal{T}$ is a set of pairs $K = \{\boldsymbol{x}_K, r_K\}$, with $\boldsymbol{x}_K \in \mathcal{P}_\mathcal{T}$ strictly inside Ω and r_K a strictly positive real number, for any $K \in \mathcal{T}$. We denote $h_K = 2r_K$.
- The set of interfaces, denoted $\mathcal{F}$, is a set of pairs $\sigma = \{\boldsymbol{x}_\sigma, \mathcal{T}_\sigma\}$, with $\boldsymbol{x}_\sigma \in \mathcal{P}_\mathcal{F}$ and $\mathcal{T}_\sigma$ a subset of $\mathcal{T}$. It is subdivided into two subsets, the set of boundary interfaces $\mathcal{F}_{ext}$ and the set of interior interfaces $\mathcal{F}_{int}$.
- The set of boundary interfaces $\mathcal{F}_{ext}$ is such that for all $K \in \mathcal{T}_\sigma$, $\boldsymbol{x}_\sigma$ is a point in $\cup_{K\in\mathcal{T}_\sigma} B(\boldsymbol{x}_K, r_K) \cap \partial\Omega$.
- The set of interior interfaces $\mathcal{F}_{int}$ is such that for all $K \in \mathcal{T}_\sigma$, $\boldsymbol{x}_\sigma$ is a point in $\cup_{K\in\mathcal{T}_\sigma} B(\boldsymbol{x}_K, r_K) \cap \mathring{\Omega}$.
- For all $(K_1, K_2) \in \mathcal{N}^2$ such that $K_1 \neq K_2$, $\boldsymbol{x}_{K_1} \neq \boldsymbol{x}_{K_2}$. For all $(\sigma_1, \sigma_2) \in \mathcal{F}^2$ such that $\sigma_1 \neq \sigma_2$, $\boldsymbol{x}_{\sigma_1} \neq \boldsymbol{x}_{\sigma_2}$.
- $\Omega \subset \bigcup_{K\in\mathcal{T}} B(\boldsymbol{x}_K, r_K)$. For any $K \in \mathcal{T}$ such that $\partial\Omega \cap \overline{B(\boldsymbol{x}_K, r_K)} \neq \emptyset$, then $\mathcal{F}_K \cap \mathcal{F}_{ext} \neq \emptyset$. For any $(K, L) \in \mathcal{T}^2$ such that $B(\boldsymbol{x}_K, r_K) \cap B(\boldsymbol{x}_L, r_L) \neq \emptyset$, then there exists $\sigma \in \mathcal{F}$ such that $(K, L) \subset \mathcal{T}_\sigma$.

For any $K \in \mathcal{T}$, we also denote $\mathcal{F}_K = \{\sigma \in \mathcal{F} \mid K \in \mathcal{T}_\sigma\}$ (the interfaces of K), which implies that for any $\sigma \in \mathcal{F}$, $\mathcal{T}_\sigma$ obviously denotes the cells connected to the interface σ and satisfies $\mathcal{T}_\sigma = \{K \in \mathcal{T} \mid \sigma \in \mathcal{F}_K\}$. The set of interfaces $\mathcal{F}$ defines the connectivity of the network and clearly depends on the chosen norm $||\boldsymbol{x}||_m^m = \sum_{i=1}^d x_i^m$, $1 \leqslant m \leqslant +\infty$

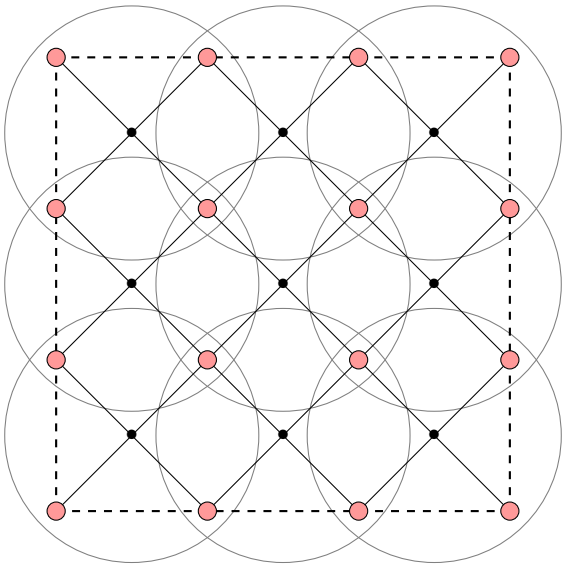

Figure 1: Example of discretization network for a square domain (light red circles are interfaces, black circles are cells, large circle represents the r_K's, lines represent the connectivity)

for computing the balls and on the size distribution r_K. In the following, we will only consider for simplicity the case $m = 2$. We denote $B_K = B(\boldsymbol{x}_K, r_K)$, and for any $x \in \mathbb{R}^d$, and we also denote

$$\mathcal{T}_{\boldsymbol{x}}^{\mathcal{N}} = \{K \in \mathcal{T} \mid \boldsymbol{x} \in B_K\} \quad \text{and} \quad \eta_{\mathcal{N}} = \sup_{x \in \mathbb{R}^d} card(\mathcal{T}_{\boldsymbol{x}}^{\mathcal{N}})$$

Finally, we denote $h = \max_{K \in \mathcal{T}} h_K$ and $\mathbb{P}_k(\mathbb{R}^d)$ the set of polynomials of order k.

All our reasoning will be based on a direct analogy with a classical mesh. We will try to approximate functions around $\boldsymbol{x}_K$ for any $K \in \mathcal{T}$, as if we had subdivided Ω in cells centered at $\boldsymbol{x}_K$. The interfaces are intended to be the analogues of the faces or vertices joining the cells, or a mixing of both. Remark that the following construction will be easier to understand if the interfaces are considered as analogous to mesh faces. By analogy with meshes it seems very natural to introduce the following admissibility condition on the discretization network:

$$\text{For all } K \in \mathcal{T},\ card(\mathcal{F}_K) \geqslant d+1 \text{ and there exists } \sigma \in \mathcal{F}_K \text{ such that } rank(\mathbb{V}_{K,\sigma}) = d \tag{1}$$

where we have denoted for all $K \in \mathcal{T}$ and all $\sigma \in \mathcal{F}_K$

$$\mathbb{V}_{K,\sigma} = (\mathbb{V}_K^{j,\sigma,\sigma'})_{1 \leqslant j \leqslant d, \sigma' \in \mathcal{F}_K} \quad \text{with} \quad \mathbb{V}_K^{j,\sigma,\sigma'} = x_{\sigma'}^j - x_\sigma^j \text{ for all } \sigma' \in \mathcal{F}_K \text{ and all } 1 \leqslant j \leqslant d \tag{2}$$

Any network satisfying (1) is of course called an admissible network. Condition (1) can be easily interpreted geometrically if we consider what would be its equivalent for a true mesh: it simply means that a cell must have at least $d+1$ true interfaces, that is $d+1$ non colinear interfaces, and is thus non degenerate. Another consequence is that for any $K \in \mathcal{T}$ the set $(\boldsymbol{x}_\sigma)_{\sigma \in \mathcal{F}_K}$ is unisolvent for first order polynomials.

1.1. Network geometry

By analogy with a classical mesh, to each cell $K \in \mathcal{T}$, we associate a strictly positive real number m_K, which is intended to be the analogue of the cell measure. To each $K \in \mathcal{T}$ and each $\sigma \in \mathcal{F}_K$, we associate a vector $\boldsymbol{\eta}_{K,\sigma}$, intended to be the analogues of the outgoing normal vector, weighted by the face measure. To circumvent some existence issues, we define a network geometry as a set of coefficients:

$$\mathcal{G} = \left((m_K)_{K\in\mathcal{T}}, (\boldsymbol{\eta}_{K,\sigma})_{K\in\mathcal{T},\sigma\in\mathcal{F}_K}, (\varepsilon_K^{0,i})_{K\in\mathcal{T},1\leqslant i\leqslant d}, (\varepsilon_K^{1,ij})_{K\in\mathcal{T},1\leqslant i,j\leqslant d}, (\varepsilon_\sigma^i)_{\sigma\in\mathcal{F}_{int},1\leqslant i\leqslant d}, \varepsilon_\Omega \right)$$

We introduce the approximate consistency properties

$$m_K > 0 \quad \text{for all } K \in \mathcal{T} \tag{3}$$

and

$$\sum_{\sigma\in\mathcal{F}_K} \eta_{K,\sigma}^i = m_K \varepsilon_K^{0,i} \quad \text{for all } K \in \mathcal{T} \text{ and all } 1 \leqslant i \leqslant d \tag{4}$$

and

$$\sum_{\sigma\in\mathcal{F}_K} \eta_{K,\sigma}^i (x_\sigma^j - x_K^j) = m_K(\delta_{ij} + \varepsilon_K^{1,ij}) \quad \text{for all } K \in \mathcal{T} \text{ and all } 1 \leqslant i, j \leqslant d \tag{5}$$

and the approximate conservation (or compatibility) properties

$$\sum_{K\in\mathcal{T}} m_K = (1 + \varepsilon_\Omega)|\Omega| \tag{6}$$

and

$$\sum_{K\in\mathcal{T}_\sigma} \eta_{K\sigma}^i = \varepsilon_\sigma^i \quad \text{for all } \sigma \in \mathcal{F}_{int} \text{ and all } 1 \leqslant i \leqslant d \tag{7}$$

We say that a network geometry is consistent if and only if it satisfies (3)-(4)-(5), and in the same way we say that it is conservative if and only if it satisfies (6)-(7). The underlying idea is very simple, the subset

$$\varepsilon(\mathcal{G}) = \left((\varepsilon_K^{0,i})_{K\in\mathcal{T},1\leqslant i\leqslant d}, (\varepsilon_K^{1,ij})_{K\in\mathcal{T},1\leqslant i,j\leqslant d}, (\varepsilon_\sigma^i)_{\sigma\in\mathcal{F}_{int},1\leqslant i\leqslant d}, \varepsilon_\Omega \right)$$

is intended to represent the approximation error we tolerate on the exact geometrical constraints. Obviously we want to have some control over it, this is why we need to introduce the constants $\theta_{\mathcal{A}} > 0$ and $p \geqslant 1$, both independent on h, and such that

$$|\varepsilon_K^{0,i}| \leqslant \theta_{\mathcal{A}} h_K^p \quad \text{for all } K \in \mathcal{T} \text{ and all } 1 \leqslant i \leqslant d \tag{8}$$

and

$$|\varepsilon_K^{1,ij}| \leqslant \theta_{\mathcal{A}} h_K^p \quad \text{for all } K \in \mathcal{T} \text{ and all } 1 \leqslant i, j \leqslant d \tag{9}$$

and

$$|\varepsilon_\Omega| \leqslant \theta_{\mathcal{A}} h^p \tag{10}$$

and

$$|\varepsilon_\sigma^i| \leqslant \theta_{\mathcal{A}} \min_{K\in\mathcal{T}_\sigma} m_K h_K^p \quad \text{for all } \sigma \in \mathcal{F}_{int} \text{ and all } 1 \leqslant i \leqslant d \tag{11}$$

Finally, we say that an approximate network geometry is admissible if and only if it is consistent and conservative. A conservative and consistent network geometry for which $\varepsilon(\mathcal{G}) = 0$ will be called an exact geometry. Notice that $\varepsilon(\mathcal{G})$ is the main difference between our notion of network geometry and those of [17, 18, 21, 22], while it also has less elements that the ones of [19, 20]. In other words, our notion of network geometry differs from those of the literature mainly by the fact that we allow it to only be approximate in terms of consistency and conservation property. Those choices are made to allow the derivation of a numerical method with a compact stencil. Another interesting consequence is the fact that contrary to most methods of the literature, the $\boldsymbol{\eta}_{K,\sigma}$ for $K \in \mathcal{T}_\sigma$ do not necessarily share the same norm or direction(as would their most natural mesh analogues $|\sigma|\boldsymbol{n}_{K,\sigma}$), even in the case of a face like connectivity (i.e. $card(\mathcal{T}_\sigma) \leqslant 2$) as it would be the case for exact geometries (see for instance [19, 20]).

In practice we will use $p = 2$ to ensure that the approximation error coming from the network geometry is smaller than the polynomial approximation error, which will become obvious when deriving consistency results for the associated numerical scheme. We keep the notation h_K^p as an easy way to distinguish the contributions of $\varepsilon(\mathcal{G})$ to the scheme's error.

Remark 1.2. Notice that if the above relations hold true then for any matrix $M \in \mathcal{M}_d(\mathbb{R})$, we have

$$\sum_{\sigma\in\mathcal{F}_K} \frac{1}{m_K}\boldsymbol{\eta}_{K,\sigma}^T M(\boldsymbol{x}_\sigma - \boldsymbol{x}_K) = tr(M) + \sum_{i=1}^d\sum_{j=1}^d \varepsilon_K^{1,ij} M_{ij} \quad \text{for all } M \in \mathcal{M}_d(\mathbb{R})$$

Applying it to a first order polynomial vector $\boldsymbol{q} = \boldsymbol{c}_q + M\boldsymbol{x}$, we easily get that

$$div\ \boldsymbol{q} + \sum_{i=1}^d\sum_{j=1}^d \varepsilon_K^{1,ij}\frac{\partial q_i}{\partial x^j} = \frac{1}{m_K}\sum_{\sigma\in\mathcal{F}_K} \boldsymbol{\eta}_{K,\sigma}^T(\boldsymbol{q}(\boldsymbol{x}_\sigma) - \boldsymbol{q}(\boldsymbol{x}_K)))$$

which is an approximate consistency result on the divergence operator as well.

1.2. Existence of network geometries

Thanks to the presence of $\varepsilon(\mathcal{G})$ in our definition of network geometries, their existence is immediate. Indeed:

Proposition 1.3. *Let $\mathcal{N}$ be an admissible network. Then there exists an admissible network geometry.*

Proof. If $\mathcal{G}$ is an admissible geometry, we have for the $\eta_{K,\sigma}$'s, the $\varepsilon_K^{1,ij}$'s, the $\boldsymbol{\varepsilon}_K^0$ and the $\boldsymbol{\varepsilon}_\sigma$

$$\left|\begin{array}{ll} \displaystyle\sum_{\sigma\in\mathcal{F}_K} \eta_{K,\sigma}^i(x_\sigma^j - x_K^j) - m_K\varepsilon_K^{1,ij} = \delta_{ij}m_K & \text{for all } K\in\mathcal{T} \text{ and all } 1\leqslant i,j\leqslant d \\ \displaystyle\sum_{\sigma\in\mathcal{F}_K} \eta_{K,\sigma}^i - m_K\varepsilon_K^{0,i} = 0 & \text{for all } K\in\mathcal{T} \text{ and all } 1\leqslant i\leqslant d \\ \displaystyle\sum_{K\in\mathcal{T}_\sigma} \eta_{K,\sigma}^i - \varepsilon_\sigma^i = 0 & \text{for all } \sigma\in\mathcal{F}_{int} \text{ and all } 1\leqslant i\leqslant d \end{array}\right.$$

Using the change of variable $\varepsilon_{m,K}^{0,i} = m_K\varepsilon_K^{0,i}$ and $\varepsilon_{m,K}^{1,ij} = m_K\varepsilon_K^{1,ij} + \eta_{K,\sigma}^i x_K^j$ and denoting

$$\mathcal{X} = \left((\boldsymbol{\eta}_{K,\sigma})_{K\in\mathcal{T},\sigma\in\mathcal{F}_K}, (\varepsilon_{m,K}^{0,i})_{K\in\mathcal{T},1\leqslant i\leqslant d}, (\varepsilon_{m,K}^{1,ij})_{K\in\mathcal{T},1\leqslant i,j\leqslant d}, (\varepsilon_\sigma^i)_{\sigma\in\mathcal{F}_{int},1\leqslant i\leqslant d}\right)$$

and $\boldsymbol{m} = (m_K)_{K\in\mathcal{T}}$, the above system can be rewritten $\mathbb{A}_G\mathcal{X} = \mathbb{L}_G\boldsymbol{m}$. Now, let us take a vector $\boldsymbol{\mu}$ in $Im(\mathbb{A}_G)$, with

$$\boldsymbol{\mu} = \left((\mu_K^{1,ij})_{K\in\mathcal{T},1\leqslant i,j\leqslant d}, (\mu_K^{0,i})_{K\in\mathcal{T},1\leqslant i\leqslant d}, (\mu_\sigma^i)_{\sigma\in\mathcal{F}_{int},1\leqslant i\leqslant d}\right).$$

We have:

$$\begin{aligned}
&(\mathbb{A}_G^T\boldsymbol{\mu})^T\mathcal{X} = \boldsymbol{\mu}^T\mathbb{A}_G\mathcal{X} \\
&= \sum_{K\in\mathcal{T}}\sum_{i=1}^d\sum_{j=1}^d\sum_{\sigma\in\mathcal{F}_K} \eta_{K,\sigma}^i x_\sigma^j\mu_K^{1,ij} + \sum_{K\in\mathcal{T}}\sum_{i=1}^d\sum_{\sigma\in\mathcal{F}_K}\eta_{K,\sigma}^i\mu_K^{0,i} + \sum_{i=1}^d\sum_{\sigma\in\mathcal{F}_{int}}\sum_{K\in\mathcal{T}_\sigma}\eta_{K,\sigma}^i\mu_\sigma^i \\
&\quad - \sum_{K\in\mathcal{T}}\sum_{i=1}^d\sum_{j=1}^d \varepsilon_{m,K}^{1,ij}\mu_K^{1,ij} - \sum_{K\in\mathcal{T}}\sum_{i=1}^d\varepsilon_{m,K}^{0,i}\mu_K^{0,i} - \sum_{i=1}^d\sum_{\sigma\in\mathcal{F}_{int}}\varepsilon_\sigma^i\mu_\sigma^i \\
&= \sum_{K\in\mathcal{T}}\sum_{i=1}^d\sum_{\sigma\in\mathcal{F}_K\cap\mathcal{F}_{int}}\eta_{K,\sigma}^i\left(\mu_\sigma^i + \mu_K^{0,i} + \sum_{j=1}^d\mu_K^{1,ij}x_\sigma^j\right) + \sum_{K\in\mathcal{T}}\sum_{i=1}^d\sum_{\sigma\in\mathcal{F}_K\cap\mathcal{F}_{ext}}\eta_{K,\sigma}^i\left(\mu_K^{0,i} + \sum_{j=1}^d\mu_K^{1,ij}x_\sigma^j\right) \\
&\quad - \sum_{K\in\mathcal{T}}\sum_{i=1}^d\sum_{j=1}^d \varepsilon_{m,K}^{1,ij}\mu_K^{1,ij} - \sum_{K\in\mathcal{T}}\sum_{i=1}^d\varepsilon_{m,K}^{0,i}\mu_K^{0,i} - \sum_{i=1}^d\sum_{\sigma\in\mathcal{F}_{int}}\varepsilon_\sigma^i\mu_\sigma^i
\end{aligned}$$

As $\mathbb{A}_G^T\boldsymbol{\mu} = 0$ is equivalent to $(\mathbb{A}_G^T\boldsymbol{\mu})^T\mathcal{X} = 0$ for all $\mathcal{X}$ we see, taking one element of $\mathcal{X}$ equal to one and all the others equal to zero, that $\mathbb{A}_G^T\boldsymbol{\mu} = 0$ immediately implies for all $1\leqslant i,j\leqslant d$ that $\mu_K^{1,ij} = 0$ and $\mu_K^{0,i} = 0$ for all $K\in\mathcal{T}$, as well as $\mu_\sigma^i = 0$ for all $\sigma\in\mathcal{F}_{int}$. Thus $\mathbb{A}_G^T\boldsymbol{\mu} = 0$ implies $\boldsymbol{\mu} = 0$. Existence is then a direct consequence of the fact that there always exists $(\varepsilon_\Omega, \boldsymbol{m})$ such that $m_K > 0$ for any $K\in\mathcal{T}$ and satisfying (6), which concludes the proof. □

The natural question that arises is whether such a result can be established for exact network geometries. In the above proof, we were able to establish existence thanks to the fact that $Ker\mathbb{A}_{\mathcal{G}}^T = \{0\}$. In the case of exact geometries the kernel of the transpose of the corresponding reduced matrix $\mathbb{A}_{\mathcal{G}}$ acting only on the $\boldsymbol{\eta}_{K,\sigma}$'s will be characterized by relations

$$0 = \mu_\sigma^i + \mu_K^{0,i} + \sum_{j=1}^{d} \mu_K^{1,ij} x_\sigma^j \text{ for all } K \in \mathcal{T} \text{ and all } \sigma \in \mathcal{F}_{int} \cap \mathcal{F}_K \text{ and } 1 \leqslant i \leqslant d \tag{12}$$

and

$$0 = \mu_K^{0,i} + \sum_{j=1}^{d} \mu_K^{1,ij} x_\sigma^j \text{ for all } K \in \mathcal{T} \text{ and all } \sigma \in \mathcal{F}_{ext} \cap \mathcal{F}_K \text{ and } 1 \leqslant i \leqslant d \tag{13}$$

and existence of exact geometries with positive cell measures will be governed by Farkas' lemma. Unfortunately, for general networks the kernel of the reduced transpose matrix will not be reduced to zero: depending on the network's connectivity, there can exist oscillating non zero vectors satisfying both (12) and (13). The consequence is that the cell measures of an exact network geometry must satisfy a potentially complex compatibility condition that can be characterized using Farkas' lemma. To ensure existence of geometries for every admissible network without computing $Ker\mathbb{A}_{\mathcal{G}}^T$ was one of the main motivation for considering augmented geometries incorporating the geometric approximation errors $\varepsilon(\mathcal{G})$. Finally, numerical experiments reveal that even on networks generated from true meshes, and thus for which existence is ensured, computing an exact geometry is a numerically difficult problem. As a slight perturbation on the point cloud is potentially enough to ensure or prevent the existence of exact geometries, our guess is that the underlying linear system remains most probably stiff even when its kernel is properly handled. The practical consequence is that constructing an exact network geometry up to machine precision can be a very costly and time consuming process. Reducing this computational cost was the second main motivation for adding $\varepsilon(\mathcal{G})$ to the usual definition of network geometries.

2. The network element method

2.1. Model problem

In order to focus on the network element method itself and to ease the understanding of the presentation, we have chosen to consider the simplest possible model problem, i.e. the Poisson problem, set on an open bounded domain Ω subset of $\mathbb{R}^d$, $d \in \mathbb{N}\backslash\{0\}$

$$-\Delta u = f \quad \text{in } \Omega, \tag{14}$$

For the sake of simplicity again, we complement it with homogeneous Dirichlet boundary conditions, i.e.

$$u = 0 \text{ on } \partial\Omega, \tag{15}$$

where $\partial\Omega = \overline{\Omega}\backslash\Omega$ is the boundary of the domain Ω, that we have assumed to be at least Lipschitz continuous, and $f \in L^2(\Omega)$. The weak solution associated to (14)-(15) is the unique $u \in H_0^1(\Omega)$ such that

$$\int_\Omega \nabla u \nabla v = \int_\Omega fv \quad \text{for all } v \text{ in } H_0^1(\Omega) \quad \Leftrightarrow \quad a(u,v) = l(v) \quad \text{for all } v \text{ in } H_0^1(\Omega) \tag{16}$$

2.2. Degrees of freedom and discrete variational formulation

Let us recall some key ideas underlying the virtual element method. Assume that we are given a mesh of Ω whose set of cells is denoted $\mathcal{T}$ to make the analogy with networks more obvious, then we can write

$$\int_\Omega \nabla u \nabla v = \sum_{K\in\mathcal{T}} \int_K \nabla u \nabla v$$

Using the virtual element projector Π_K^{VEM} onto first order polynomial functions, which is exact when tested again first order polynomial functions (see [2]) , one has

$$\int_\Omega \nabla u \nabla v = \sum_{K\in\mathcal{T}} \int_K \nabla \Pi_K^{VEM}(u) \nabla \Pi_K^{VEM}(v) + \sum_{K\in\mathcal{T}} \int_K (\nabla u - \nabla \Pi_K^{VEM}(u))(\nabla v - \nabla \Pi_K^{VEM}(v))$$

The first term in the sum is the consistency part, which is used under this form in practice, and is intended to account for the polynomial part of the discrete approximation. For the second term, the idea of the virtual element method is that it can be replaced by a generic stabilization bilinear form

$$s_K^{VEM}(u - \Pi_K^{VEM}(u), v - \Pi_K^{VEM}(v))$$

which only needs to scale with h_K in the same way than the term it replaces, leading to the discrete variational formulation for the first order virtual element method (denoting π_k the L^2 projection on polynomials of order k)

$$\sum_{K\in\mathcal{T}} \int_K \nabla\Pi_K^{VEM}(u)\nabla\Pi_K^{VEM}(v) + \sum_{K\in\mathcal{T}} s_K^{VEM}(u - \Pi_K^{VEM}(u), v - \Pi_K^{VEM}(v)) = \sum_{K\in\mathcal{T}} \int_K f\pi_0(v)$$

To construct the network element method, we will follow the principles of the virtual element method, but using our discretization network rather than a mesh. It means that we will construct a discrete bilinear form based on the discretization network and its associated network geometry that will handle the polynomial part, in some sense, of our degrees of freedom. Doing so, we will need to relax some of the consistency requirements of the mimetic approach to handle the fact that our network geometries are only approximate. Then, we complete it with a stabilization term that, roughly speaking, just need to have the correct scaling and to approximately vanish for polynomial degrees of freedom. As mentioned in the introduction, this clear separation between the consistency part acting on polynomials and the stabilization part is also very reminiscent of discontinuous Galerkin methods ([14]).

Let $\mathcal{N}$ be an admissible discretization network, endowed with an admissible network geometry $\mathcal{G}$. We associate to $\mathcal{N}$ the following space of degrees of freedom

$$X_{\mathcal{N}} = \left\{(u_\sigma)_{\sigma\in\mathcal{F}} \mid u_\sigma \in \mathbb{R} \text{ for all } \sigma \in \mathcal{F}\right\}$$

and denote $\boldsymbol{U} = (u_\sigma)_{\sigma\in\mathcal{F}}$. The idea behind those degrees of freedom is simply to mimic the VEM, assigning degrees of freedom to interfaces between cells which in the case of the first order VEM means assigning degrees of freedom to mesh vertices. Notice that choice leads to a space of degrees of freedom very similar to those of the meshless methods of [19, 20]: we want to compute an approximation around the points of $\mathcal{T}$, but our degrees of freedom are located at the interfaces between elements of $\mathcal{T}$. To account for homogeneous Dirichlet boundary conditions, we also consider

$$X_{\mathcal{N},0} = \{\boldsymbol{U} \in X_{\mathcal{N}} \mid u_\sigma = 0 \text{ for all } \sigma \in \mathcal{F}_{ext}\}$$

and we define the local set of degrees of freedom associated to a cell by

$$X_{\mathcal{N},K} = \left\{(u_\sigma)_{\sigma\in\mathcal{F}_K} \mid u_\sigma \in \mathbb{R} \text{ for all } \sigma \in \mathcal{F}_K\right\}$$

Of course, for any $\boldsymbol{U} \in X_{\mathcal{N}}$, we denote $\boldsymbol{U}_K = (u_\sigma)_{\sigma\in\mathcal{F}_K}$. Let us now describe the derivation of the counterpart of the virtual element projector in our context. To any cell $K \in \mathcal{T}$, we associate a point $\overline{\boldsymbol{x}}_K$ and denote

$$\overline{\boldsymbol{x}}_K = \sum_{\sigma\in\mathcal{F}_K} \gamma_{K,\sigma}\boldsymbol{x}_\sigma \quad \text{where} \quad \sum_{\sigma\in\mathcal{F}_K} \gamma_{K,\sigma} = 1$$

thus the $(\gamma_{K,\sigma})_{\sigma\in\mathcal{F}_K}$ forms a barycentric interpolation for $\overline{\boldsymbol{x}}_K$ from the interface points $(\boldsymbol{x}_\sigma)_{\sigma\in\mathcal{F}_K}$. Let us denote

$$\mathcal{M}_K(\boldsymbol{U}_K) = \sum_{\sigma\in\mathcal{F}_K} \gamma_{K,\sigma}u_\sigma$$

To any cell $K \in \mathcal{T}$, we associate the local reconstruction operator Π_K defined by

$$\left|\begin{array}{rcl} \Pi_K : X_{\mathcal{N},K} & \longmapsto & \mathbb{P}_1(\mathbb{R}^d) \\ \boldsymbol{U}_K & \longrightarrow & \Pi_K(\boldsymbol{U}_K) = \mathcal{M}_K(\boldsymbol{U}_K) + \nabla_K(\boldsymbol{U}_K)\cdot(\boldsymbol{x} - \overline{\boldsymbol{x}}_K) \end{array}\right. \tag{17}$$

where

$$\left|\begin{array}{rcl} \nabla_K : X_{\mathcal{N},K} & \longmapsto & \mathbb{P}_0(\mathbb{R}^d)^d \\ \boldsymbol{U}_K & \longrightarrow & \nabla_K(\boldsymbol{U}_K) = \dfrac{1}{m_K}\displaystyle\sum_{\sigma\in\mathcal{F}_K} u_\sigma\boldsymbol{\eta}_{K,\sigma} \end{array}\right. \tag{18}$$

Notice that the above definition implies that

$$\mathcal{M}_K\left(\left(\Pi_K\left(\boldsymbol{U}_K\right)\left(\boldsymbol{x}_\sigma\right)\right)_{\sigma\in\mathcal{F}_K}\right) \equiv \sum_{\sigma\in\mathcal{F}_K} \gamma_{K,\sigma}\Pi_K\left(\boldsymbol{U}_K\right)\left(\boldsymbol{x}_\sigma\right) = \mathcal{M}_K(\boldsymbol{U}_K) \quad \text{and} \quad \nabla\Pi_K\left(\boldsymbol{U}_K\right) = \nabla_K\left(\boldsymbol{U}_K\right) \tag{19}$$

Finally, we extend the definition of Π_K to all $X_{\mathcal{N}}$ by setting

$$\left|\begin{array}{rcl} \Pi_K : X_{\mathcal{N}} & \longmapsto & \mathbb{P}_1(\mathbb{R}^d) \\ \boldsymbol{U} & \longrightarrow & \Pi_K\left(\boldsymbol{U}_K\right) \end{array}\right. \tag{20}$$

and we do the same for $\mathcal{M}_K$ and ∇_K. For any $\boldsymbol{U} \in X_{\mathcal{N}}$, if the geometry is exact then as its VEM equivalent $\Pi_K\left(\boldsymbol{U}_K\right)$ corresponds to the local polynomial part of $\boldsymbol{U}$ around $\boldsymbol{x}_K$. Indeed, an immediate consequence of (4)-(5) is that if $\boldsymbol{U}_{\pi,K}$ is the set of degrees of freedom of a polynomial $\pi_K \in \mathbb{P}_1(\mathbb{R}^d)$, then $\Pi_K\left(\boldsymbol{U}_{\pi,K}\right) = \pi_K$, while if $\boldsymbol{U}_{\pi,K}$ is the set of degrees of freedom of a polynomial $\pi_K \in \mathbb{P}_0(\mathbb{R}^d)$, then $\mathcal{M}_K\left(\boldsymbol{U}_{\pi,K}\right) = \pi_K$. However in the more general case of an approximate geometry, we only get an approximation of the polynomial part and the above formulae are only true up to a term of magnitude h_K^p. In other words, it means that our discrete reconstruction $\Pi_K\left(\boldsymbol{U}_{\pi,K}\right)$ will not satisfy exactly what engineers call the patch test. Numerical results will illustrate that this approximate consistency is nevertheless enough to obtain convergence and that satisfying exactly the patch test is not mandatory.

For any $\varphi \in C^0(\mathbb{R}^d)$ (and more generally for any function for which it makes sense), we denote $\mathcal{D}_K(\varphi) = \left(\varphi(\boldsymbol{x}_\sigma)\right)_{\sigma\in\mathcal{F}_K}$ the local set of degrees of freedom associated with φ, while $\mathcal{D}(\varphi) = \left(\varphi(\boldsymbol{x}_\sigma)\right)_{\sigma\in\mathcal{F}}$ denotes the complete set of degrees of freedom associated with φ.

We then define $a_h^K : X_{\mathcal{N},K} \times X_{\mathcal{N},K} \longmapsto \mathbb{R}$ by

$$a_h^K(\boldsymbol{U}_K, \boldsymbol{V}_K) = m_K \nabla\Pi_K\left(\boldsymbol{U}_K\right) \cdot \nabla\Pi_K\left(\boldsymbol{V}_K\right) + s^K\left(\boldsymbol{U}_K - \mathcal{D}_K\left(\Pi_K\left(\boldsymbol{U}_K\right)\right), \boldsymbol{V}_K - \mathcal{D}_K\left(\Pi_K\left(\boldsymbol{V}_K\right)\right)\right) \tag{21}$$

where s^K is a positive symmetric bilinear form on $X_{\mathcal{N},K} \times X_{\mathcal{N},K}$, such that

$$s^K(\boldsymbol{U}_K, \boldsymbol{V}_K) = m_K h_K^{-2} \sum_{\sigma\in\mathcal{F}_K} \sum_{\sigma'\in\mathcal{F}_K} S_{K,\sigma,\sigma'} u_\sigma v_{\sigma'} \tag{22}$$

where $S_K = \left(S_{K,\sigma,\sigma'}\right)_{\sigma,\sigma'\in\mathcal{F}_K}$ can be any symmetric positive definite matrix independent on the geometry $\mathcal{G}$ associated to the network, for which we denote

$$S_* = \inf_{K\in\mathcal{T}} \inf_{\boldsymbol{\xi}\in\mathbb{R}^{card(\mathcal{F}_K)}, ||\boldsymbol{\xi}||=1} \boldsymbol{\xi}^T S_K \boldsymbol{\xi} \quad \text{and} \quad S^* = \sup_{K\in\mathcal{T}} \sup_{\boldsymbol{\xi}\in\mathbb{R}^{card(\mathcal{F}_K)}, ||\boldsymbol{\xi}||=1} \boldsymbol{\xi}^T S_K \boldsymbol{\xi}$$

Of course, we define a bilinear form $a_h : X_{\mathcal{N}} \times X_{\mathcal{N}} \longmapsto \mathbb{R}$ by setting

$$a_h(\boldsymbol{U}, \boldsymbol{V}) = \sum_{K\in\mathcal{T}} a_h^K(\boldsymbol{U}_K, \boldsymbol{V}_K)$$

For the right-hand side, assume that f_K is an approximation of f at $\overline{\boldsymbol{x}}_K$ (for instance, one can use $f(\overline{\boldsymbol{x}}_K)$ if f is regular enough for this quantity to make sense), then we define a linear form $l_h : X_{\mathcal{N}} \longmapsto \mathbb{R}$ by setting

$$l_h(\boldsymbol{V}) = \sum_{K\in\mathcal{T}} m_K f_K \mathcal{M}_K(\boldsymbol{V}_K)$$

Then, the discretization by the network element method consists in finding a solution $\boldsymbol{U} \in X_{\mathcal{N},0}$ of

$$a_h(\boldsymbol{U}, \boldsymbol{V}) = l_h(\boldsymbol{V}) \quad \text{for all } \boldsymbol{V} \in X_{\mathcal{N},0} \tag{23}$$

2.3. Some practical remarks on the network element method

Let us make the above discrete variational formulation more explicit in terms of degrees of freedom. For any $(\boldsymbol{U},\boldsymbol{V}) \in X_{\mathcal{N}}^2$

$$a_h^K(\boldsymbol{U}_K,\boldsymbol{V}_K) = m_K \nabla\Pi_K(\boldsymbol{U})\nabla\Pi_K(\boldsymbol{V}) + m_K h_K^{-2} \sum_{\sigma\in\mathcal{F}_K}\sum_{\sigma'\in\mathcal{F}_K} S_{K,\sigma,\sigma'}(u_\sigma - \Pi_K(\boldsymbol{U})(\boldsymbol{x}_\sigma))(v_{\sigma'} - \Pi_K(\boldsymbol{V})(\boldsymbol{x}_{\sigma'}))$$

Notice that by definition of Π_K, we have

$$\begin{aligned} u_\sigma - \Pi_K(\boldsymbol{U})(\boldsymbol{x}_\sigma) &= u_\sigma - \mathcal{M}_K(\boldsymbol{U}) - \nabla_K(\boldsymbol{U})\cdot(\boldsymbol{x}_\sigma - \overline{\boldsymbol{x}}_K) \\ &= u_\sigma - \sum_{\sigma''\in\mathcal{F}_K} \gamma_{K,\sigma''} u_{\sigma''} - \frac{1}{m_K}\sum_{\sigma''\in\mathcal{F}_K} u_{\sigma''}\boldsymbol{\eta}_{K,\sigma''}\cdot(\boldsymbol{x}_\sigma - \overline{\boldsymbol{x}}_K) \end{aligned}$$

A straightforward computation leads to

$$a_h^K(\boldsymbol{U}_K,\boldsymbol{V}_K) = \sum_{\sigma\in\mathcal{F}_K}\sum_{\sigma'\in\mathcal{F}_K} \mathbb{A}_K^{\sigma,\sigma'} u_\sigma v_{\sigma'}$$

where

$$\left|\begin{array}{l} \mathbb{A}_K^{\sigma,\sigma'} = \dfrac{1}{m_K}\Lambda_K\boldsymbol{\eta}_{K,\sigma}\cdot\boldsymbol{\eta}_{K,\sigma'} + m_K h_K^{-2}\mathbb{S}_K^{\sigma,\sigma'}, \\ \mathbb{S}_K^{\sigma,\sigma'} = y_{K,\sigma'}^T S_K y_{K,\sigma}, \\ y_{K,\sigma} = (y_{K,\sigma}^{\sigma''})_{\sigma''\in\mathcal{F}_K}, \quad y_{K,\sigma}^{\sigma''} = \delta_{\sigma,\sigma''} - \gamma_{K,\sigma} - \dfrac{1}{m_K}\boldsymbol{\eta}_{K,\sigma}\cdot(\boldsymbol{x}_{\sigma''} - \boldsymbol{x}_K) \end{array}\right. \tag{24}$$

where $\delta_{\sigma,\sigma''}$ is the Kronecker symbol. For the second member, we immediately get

$$l_h(\boldsymbol{V}) = \sum_{K\in\mathcal{T}}\sum_{\sigma\in\mathcal{F}_K} \mathbb{L}_K^\sigma v_\sigma \quad \text{with} \quad \mathbb{L}_K^\sigma = m_K f_K \gamma_{K,\sigma}$$

and thus (23) is equivalent to solving

$$\sum_{K\in\mathcal{T}}\sum_{\sigma\in\mathcal{F}_K}\sum_{\sigma'\in\mathcal{F}_K} \mathbb{A}_K^{\sigma,\sigma'} u_\sigma v_{\sigma'} = \sum_{K\in\mathcal{T}}\sum_{\sigma\in\mathcal{F}_K} \mathbb{L}_K^\sigma v_\sigma \tag{25}$$

complemented by the boundary conditions $u_\sigma = 0$ and $v_\sigma = 0$ for any $\sigma \in \mathcal{F}_{ext}$. To get an even more familiar version of this system, let us introduce the square matrix $\mathbb{A}$ of size $card(\mathcal{F})$ as well as $\boldsymbol{F} \in \boldsymbol{X}_{\mathcal{N}}$ such that and for any $\sigma,\sigma' \in \mathcal{F}_{int}$

$$\mathbb{A}_{\sigma',\sigma} = \sum_{K\in\mathcal{T}_\sigma\cap\mathcal{T}_{\sigma'}} \mathbb{A}_K^{\sigma,\sigma'} \quad \text{and} \quad \boldsymbol{f}_\sigma = \sum_{K\in\mathcal{T}_\sigma} \mathbb{L}_K^\sigma$$

and for any $\sigma,\sigma' \in \mathcal{F}_{int}$

$$\mathbb{A}_{\sigma',\sigma} = \delta_{\sigma',\sigma} \quad \text{and} \quad f_\sigma = 0$$

Problem (23) is then equivalent to solving the linear system

$$\mathbb{A}\boldsymbol{U} = \boldsymbol{F}$$

Then, assuming an ordering of the interfaces such that the elements of $\mathcal{F}_{ext}$ are the last ones in the list, this rewrites

$$\begin{pmatrix} \mathbb{A}_{\mathcal{F}_{int},\mathcal{F}_{int}} & 0 \\ 0 & \mathbb{I}_{\mathcal{F}_{ext}} \end{pmatrix} \boldsymbol{U} = \begin{pmatrix} \boldsymbol{F}_{\mathcal{F}_{int}} \\ \boldsymbol{0} \end{pmatrix}$$

Algorithm 1 Pseudo code for assembling the network element method

Initialization:
All matrix and vector coefficients are set to zero.
Set one on the diagonal entries of the matrix corresponding to elements of $\mathcal{F}_{ext}$
Assembly loop:
for $K \in \mathcal{T}$ **do**
  Compute the barycentric weights $\gamma_{K,\sigma}$ if necessary
  Compute all the local terms $\mathbb{A}_K^{\sigma,\sigma'}$
  for $\sigma, \sigma' \in \mathcal{F}_K$ **do**
    if both σ and $\sigma' \in \mathcal{F}_{int}$, add $\mathbb{A}_K^{\sigma,\sigma'}$ to the line σ' of the matrix $\mathbb{A}$ at column σ
  end for
  for $\sigma \in \mathcal{F}_K$ **do**
    if $\sigma \in \mathcal{F}_{int}$, add $\mathbb{L}_K^{\sigma}$ to the line σ of the second member
  end for
end for

where $\mathbb{I}_{\mathcal{F}_{ext}}$ is the identity matrix on $\mathbb{R}^{card(\mathcal{F}_{ext})}$, which is the familiar form of a finite element matrix. The boundary degrees of freedom can then be either eliminated or handled through Schur's complement to reduce the size of the final linear system. Also notice that the obtained matrix has the same kind of sparsity pattern than finite element matrices, as given an interface σ its stencil is reduced to the interfaces sharing a common cell. The scheme can thus be implemented in a finite element fashion, looping over all cells $K \in \mathcal{T}$ and incrementally constructing the global system. In other words, a possible implementation would be to follow the pseudo code described in algorithm 1. Due to the fact the discretization network has a mesh-like structure and that we can resort to classical finite element assembling procedures, the network element method can relatively easily be incorporated in existing codes already dealing with unstructured meshes.

2.4. Basic properties of the network element method

Obviously for the above method to be of any practical interest, we must check existence, uniqueness and stability of its solutions, as well as its consistency with the continuous problem. To this end, we introduce the first measures of quality of the discretization network

$$\theta_\Pi = \sup_{K\in\mathcal{T}} \sup_{\sigma\in\mathcal{F}_K} h_K \left| \frac{\boldsymbol{\eta}_{K,\sigma}}{m_K} \right| \quad \text{and} \quad \theta_{\mathcal{M}} = \sup_{K\in\mathcal{T}} \sup_{\sigma\in\mathcal{F}_K} |\gamma_{K,\sigma}|$$

It is clear that the smaller θ_Π and $\theta_{\mathcal{M}}$ are, the better the network is. Let us also denote

$$\theta_{\mathcal{F}} = \max_{K\in\mathcal{T}} card(\mathcal{F}_K)$$

We start by considering the consistency of the method with the original problem. Fortunately, it is an immediate consequence of the consistency of the approximate network geometry.

Lemma 2.1 (Strong consistency)**.** *Let $(\mathcal{N},\mathcal{G})$ be an admissible discretization network and an associated admissible network geometry. For any $\varphi \in C_c^1(\mathbb{R}^d)$, there exists $C_\varphi > 0$ depending only on φ such that for any $K \in \mathcal{T}$ and any $\boldsymbol{x} \in B(\boldsymbol{x}_K, \xi_K)$ where $\xi_K \leqslant \kappa_\xi r_K$ with $\kappa_\xi \geqslant 1$*

$$|\varphi(\boldsymbol{x}) - \mathcal{M}_K(\mathcal{D}_K(\varphi))| \leqslant d\kappa_\xi \theta_{\mathcal{M}} \theta_{\mathcal{F}} C_\varphi h_K$$

while for any $\varphi \in C_c^2(\mathbb{R}^d)$, there exists another $C_\varphi > 0$ depending only on φ such that for any $K \in \mathcal{T}$ and any $\boldsymbol{x} \in B(\boldsymbol{x}_K, \xi_K)$

$$|\nabla\varphi(\boldsymbol{x}) - \nabla_K(\mathcal{D}_K(\varphi))| \leqslant \theta_{\mathcal{A}}(d^{1/2} + d^{3/2} + d\kappa_\xi h_K) C_\varphi h_K^p + {\kappa_\xi}^2 \theta_\Pi \theta_{\mathcal{F}} C_\varphi h_K$$

and

$$|\varphi(\boldsymbol{x}) - \Pi_K(\mathcal{D}_K(\varphi))| \leqslant \frac{1}{2}\kappa_\xi\theta_{\mathcal{M}}\theta_{\mathcal{F}}\theta_{\mathcal{A}}(d^{1/2} + d^{3/2} + d\kappa_\xi h_K)C_\varphi h_K^{p+1} + {\kappa_\xi}^2\theta_{\mathcal{M}}\theta_{\mathcal{F}}(1 + \frac{1}{2}\kappa_\xi\theta_\Pi\theta_{\mathcal{F}})C_\varphi h_K^2$$

For any $\Phi \in C_c^2(\mathbb{R}^d)^d$, there exists $C_\Phi > 0$ depending only on Φ such that, for any $K \in \mathcal{T}$ and any $\boldsymbol{x} \in B(\boldsymbol{x}_K, \xi_K)$

$$|div\ \Phi(\boldsymbol{x}) - \mathcal{DIV}_K(\mathcal{D}_K(\Phi))| \leqslant \theta_{\mathcal{A}}(d + d^2 + d^{3/2}\kappa_\xi h_K)C_\Phi h_K^p + {\kappa_\xi}^2\theta_\Pi\theta_{\mathcal{F}}C_\Phi h_K$$

where

$$\mathcal{DIV}_K(\mathcal{D}_K(\Phi)) = \frac{1}{m_K}\sum_{\sigma\in\mathcal{F}_K}\boldsymbol{\eta}_{K,\sigma}^T\Phi(\boldsymbol{x}_\sigma)$$

Proof. Using Taylor's expansion formula, we have for any $\sigma \in \mathcal{F}_K$

$$\varphi(\boldsymbol{x}_\sigma) = \varphi(\boldsymbol{x}) + \sum_{|\alpha|=1}(\boldsymbol{x}_\sigma - \boldsymbol{x})^\alpha\int_0^1\partial^\alpha\varphi(\boldsymbol{x} + t(\boldsymbol{x}_\sigma - \boldsymbol{x}))$$

and the first result follows by noticing that

$$\left|\sum_{|\alpha|=1}(\boldsymbol{x}_\sigma - \boldsymbol{x})^\alpha\int_0^1\partial^\alpha\varphi(\boldsymbol{x} + t(\boldsymbol{x}_\sigma - \boldsymbol{x}))\right| \leqslant d\kappa_\xi h_K\sup_{|\alpha|=1}||\partial^\alpha\varphi||_{L^\infty(\overline{\Omega})}$$

as $|\boldsymbol{x}_\sigma - \boldsymbol{x}| \leqslant r_K + \xi_K \leqslant \kappa_\xi h_K$ for any $\boldsymbol{x} \in B(\boldsymbol{x}_K, \xi_K)$, and using the fact that $\sum_{\sigma\in\mathcal{F}_K}\gamma_{K,\sigma} = 1$. For the second result, using again Taylor's expansion formula, we have for any $\sigma \in \mathcal{F}_K$

$$\varphi(\boldsymbol{x}_\sigma) = \varphi(\boldsymbol{x}) + \nabla\varphi(\boldsymbol{x})\cdot(\boldsymbol{x}_\sigma - \boldsymbol{x}) + 2\sum_{|\alpha|=2}\frac{(\boldsymbol{x}_\sigma - \boldsymbol{x})^\alpha}{\alpha!}\int_0^1(1-t)\partial^\alpha\varphi(\boldsymbol{x} + t(\boldsymbol{x}_\sigma - \boldsymbol{x}))$$

and thus, by definition

$$\nabla_K(\mathcal{D}_K(\varphi)) = \sum_{\sigma\in\mathcal{F}_K}\frac{1}{m_K}\varphi(\boldsymbol{x})\boldsymbol{\eta}_{K,\sigma} + \sum_{\sigma\in\mathcal{F}_K}\frac{1}{m_K}\nabla\varphi(\boldsymbol{x})\cdot(\boldsymbol{x}_K - \boldsymbol{x})\boldsymbol{\eta}_{K,\sigma}$$

$$+\sum_{\sigma\in\mathcal{F}_K}\frac{1}{m_K}\nabla\varphi(\boldsymbol{x})\cdot(\boldsymbol{x}_\sigma - \boldsymbol{x}_K)\boldsymbol{\eta}_{K,\sigma} + \frac{2}{m_K}\sum_{\sigma\in\mathcal{F}_K}\left(\sum_{|\alpha|=2}\frac{(\boldsymbol{x}_\sigma - \boldsymbol{x})^\alpha}{\alpha!}\int_0^1(1-t)\partial^\alpha\varphi(\boldsymbol{x} + t(\boldsymbol{x}_\sigma - \boldsymbol{x}))\right)\boldsymbol{\eta}_{K,\sigma}$$

Using the first order approximate consistency properties (4) and (5), this leads to

$$\nabla_K(\mathcal{D}_K(\varphi)) = \nabla\varphi(\boldsymbol{x}) + (\varphi(\boldsymbol{x}) + \nabla\varphi(\boldsymbol{x})\cdot(\boldsymbol{x}_K - \boldsymbol{x})))\,\boldsymbol{\varepsilon}_K^0 + \sum_{i=1}^d\sum_{j=1}^d\boldsymbol{\varepsilon}_K^{1,ij}\partial_{x_j}\varphi(\boldsymbol{x})\boldsymbol{e}_i$$

$$+\frac{2}{m_K}\sum_{\sigma\in\mathcal{F}_K}\left(\sum_{|\alpha|=2}\frac{(\boldsymbol{x}_\sigma - \boldsymbol{x})^\alpha}{\alpha!}\int_0^1(1-t)\partial^\alpha\varphi(\boldsymbol{x} + t(\boldsymbol{x}_\sigma - \boldsymbol{x}))\right)\boldsymbol{\eta}_{K,\sigma}$$

Noticing that

$$\left|2\sum_{|\alpha|=2}\frac{(\boldsymbol{x}_\sigma - \boldsymbol{x})^\alpha}{\alpha!}\int_0^1(1-t)\partial^\alpha\varphi(\boldsymbol{x} + t(\boldsymbol{x}_\sigma - \boldsymbol{x}))\right| \leqslant \left(2\sum_{|\alpha|=2}\frac{(\boldsymbol{x}_\sigma - \boldsymbol{x})^\alpha}{\alpha!}\right)\sup_{|\alpha|=2}||\partial^\alpha\varphi||_{L^\infty(\mathbb{R}^d)}$$

the multinomial theorem

$$2\sum_{|\alpha|=2}\frac{(\boldsymbol{x}_\sigma-\boldsymbol{x})^\alpha}{\alpha!}=|\boldsymbol{x}_\sigma-\boldsymbol{x}|^2$$

and the fact that $|\boldsymbol{x}_\sigma-\boldsymbol{x}|^2\leqslant {\kappa_\xi}^2 h_K^2$ for any $\boldsymbol{x}\in B(\boldsymbol{x}_K,\xi_K)$, we immediately get

$$|\nabla\varphi(\boldsymbol{x})-\nabla_K(\mathcal{D}_K(\varphi))|\leqslant\theta_{\mathcal{A}}h_K^p(d^{1/2}||\varphi||_{L^\infty(\mathbb{R}^d)}+(d^{3/2}+d\kappa_\xi h_K)\sup_{|\alpha|=1}||\partial^\alpha\varphi||_{L^\infty(\mathbb{R}^d)})$$

$$+{\kappa_\xi}^2\theta_\Pi\theta_{\mathcal{F}}h_K\sup_{|\alpha|=2}||\partial^\alpha\varphi||_{L^\infty(\mathbb{R}^d)}$$

Noticing that

$$\Pi_K(\mathcal{D}_K(\varphi))=\mathcal{M}_K(\mathcal{D}_K(\varphi))+\nabla_K(\mathcal{D}_K(\varphi))\cdot(\boldsymbol{x}-\overline{\boldsymbol{x}}_K)$$

and that the same Taylor's expansion leads to

$$\mathcal{M}_K(\mathcal{D}_K(\varphi))=\varphi(\boldsymbol{x})+\nabla\varphi(\boldsymbol{x})\cdot(\overline{\boldsymbol{x}}_K-\boldsymbol{x})+2\sum_{\sigma\in\mathcal{F}_K}\gamma_{K,\sigma}\sum_{|\alpha|=2}\frac{(\boldsymbol{x}_\sigma-\boldsymbol{x})^\alpha}{\alpha!}\int_0^1(1-t)\partial^\alpha\varphi(\boldsymbol{x}+t(\boldsymbol{x}_\sigma-\boldsymbol{x}))$$

Using the multinomial theorem as above, for any $\boldsymbol{x}\in B(\boldsymbol{x}_K,\xi_K)$ the last term is easily bounded by

$$\left|2\sum_{\sigma\in\mathcal{F}_K}\gamma_{K,\sigma}\sum_{|\alpha|=2}\frac{(\boldsymbol{x}_\sigma-\boldsymbol{x})^\alpha}{\alpha!}\int_0^1(1-t)\partial^\alpha\varphi(\boldsymbol{x}+t(\boldsymbol{x}_\sigma-\boldsymbol{x}))\right|\leqslant\theta_{\mathcal{M}}\theta_{\mathcal{F}}|\boldsymbol{x}_\sigma-\boldsymbol{x}|^2\sup_{|\alpha|=2}||\partial^\alpha\varphi||_{L^\infty(\mathbb{R}^d)}$$

while by construction we have that for any $\boldsymbol{x}\in B(\boldsymbol{x}_K,\xi_K)$

$$|\overline{\boldsymbol{x}}_K-\boldsymbol{x}|\leqslant\left|\sum_{\sigma\in\mathcal{F}_K}\gamma_{K,\sigma}(\boldsymbol{x}_\sigma-\boldsymbol{x})\right|\leqslant\theta_{\mathcal{M}}\sum_{\sigma\in\mathcal{F}_K}|\boldsymbol{x}_\sigma-\boldsymbol{x}|\leqslant\theta_{\mathcal{M}}\theta_{\mathcal{F}}(r_K+\xi_K)\leqslant\kappa_\xi\theta_{\mathcal{M}}\theta_{\mathcal{F}}h_K$$

as $\sum_{\sigma\in\mathcal{F}_K}\gamma_{K,\sigma}=1$, which, combined with the previous result, gives for any $\boldsymbol{x}\in B(\boldsymbol{x}_K,\xi_K)$

$$|(\nabla\varphi(\boldsymbol{x})-\nabla_K(\mathcal{D}_K(\varphi)))\cdot(\overline{\boldsymbol{x}}_K-\boldsymbol{x})|\leqslant\frac{1}{2}\kappa_\xi\theta_{\mathcal{M}}\theta_{\mathcal{F}}\theta_{\mathcal{A}}(d^{1/2}+d^{3/2}+d\kappa_\xi h_K)C_\varphi h_K^{p+1}$$

$$+\frac{1}{2}{\kappa_\xi}^3\theta_{\mathcal{M}}\theta_\Pi\theta_{\mathcal{F}}^2C_\varphi h_K^2$$

Combining those two estimates immediately gives the third result. The last result on divergence operators can be established proceeding the same way applying Taylor's expansion formula on each component of Φ and the first order approximate consistency properties. □

Remark 2.2. In general, what we want to establish on a given numerical scheme is consistency at some h^q approximation order, for any functions belonging to a given space V. To do so, in general one uses the fact that $\mathbb{P}_q(\mathbb{R}^d)\subset V$. Then, if polynomials are exactly computed by the numerical scheme (i.e. if the method satisfies the patch test exactly), combined with polynomial approximation results this is in general enough to establish the consistency of the scheme. However, this approach gives a very specific role to polynomials. Indeed, if a method approximates polynomials with the expected rate h^q, then polynomial approximation results will also lead to the consistency of the method and exactness on polynomials is not necessary. Thus, allowing approximate geometries is definitely not a major issue provided we have the correct amount of control over the additional consistency errors, reflected in our consistency estimates by some extra terms in h_K^p with respect to what one could expect.

Thanks to the variational nature of the network element method, its stability analysis can be conducted in a classical Hilbertian setting. To do so, we naturally endow the space of degrees of freedom $X_{\mathcal{N},K}$ with the bilinear forms

$$(\boldsymbol{U},\boldsymbol{V})_{0,K} = m_K \mathcal{M}_K(\boldsymbol{U}_K)\,\mathcal{M}_K(\boldsymbol{V}_K) + \sum_{\sigma\in\mathcal{F}_K} m_K (u_\sigma - \mathcal{M}_K(\boldsymbol{U}_K))(v_\sigma - \mathcal{M}_K(\boldsymbol{V}_K))$$

and

$$(\boldsymbol{U},\boldsymbol{V})_{1,K} = \sum_{\sigma\in\mathcal{F}_K} m_K h_K^{-2} (u_\sigma - \mathcal{M}_K(\boldsymbol{U}_K))(v_\sigma - \mathcal{M}_K(\boldsymbol{V}_K))$$

and the associated norm $||\boldsymbol{U}||_{0,K}^2 = (\boldsymbol{U},\boldsymbol{U})_{0,K}$ and semi-norm $|\boldsymbol{U}|_{1,K}^2 = (\boldsymbol{U},\boldsymbol{U})_{1,K}$, while we denote $||\boldsymbol{U}||_{X,K}^2 = ||\boldsymbol{U}||_{0,K}^2 + |\boldsymbol{U}|_{1,K}^2$. We endow the space of degrees of freedom $X_{\mathcal{N}}$ with the bilinear forms

$$(\boldsymbol{U},\boldsymbol{V})_0 = \sum_{K\in\mathcal{T}} (\boldsymbol{U},\boldsymbol{V})_{0,K} \quad \text{and} \quad (\boldsymbol{U},\boldsymbol{V})_1 = \sum_{K\in\mathcal{T}} (\boldsymbol{U},\boldsymbol{V})_{1,K}$$

and the associated norm $||\boldsymbol{U}||_0^2 = (\boldsymbol{U},\boldsymbol{U})_0$ and semi-norm $|\boldsymbol{U}|_1^2 = (\boldsymbol{U},\boldsymbol{U})_1$. Then we define

$$(\boldsymbol{U},\boldsymbol{V})_X = (\boldsymbol{U},\boldsymbol{V})_0 + (\boldsymbol{U},\boldsymbol{V})_1 \quad \text{and} \quad ||\boldsymbol{U}||_X^2 = (\boldsymbol{U},\boldsymbol{U})_X$$

which are obviously a scalar product and the associated norm on $X_{\mathcal{N}}$, making $X_{\mathcal{N}}$ a Hilbert space. To conclude this preliminary study of the network element method, it remains to study the stability properties of the bilinear and linear forms defined above regarding those norms.

Lemma 2.3. *For any $\boldsymbol{U}_K \in X_{\mathcal{N},K}$:*

$$m_K \left|\nabla_K(\boldsymbol{U}_K)\right|^2 \leqslant 2\theta_\Pi^2\theta_{\mathcal{F}} \sum_{\sigma\in\mathcal{F}_K} m_K h_K^{-2}(u_\sigma - \mathcal{M}_K(\boldsymbol{U}_K))^2 + 2d m_K^2 \theta_{\mathcal{A}}^2 h_K^{2p} |\mathcal{M}_K(\boldsymbol{U}_K)|^2$$

Proof. Using the conditions of (4)-(5), we can rewrite

$$\nabla_K(\boldsymbol{U}_K) = \frac{1}{m_K} \sum_{\sigma\in\mathcal{F}_K} (u_\sigma - \mathcal{M}_K(\boldsymbol{U}_K))\boldsymbol{\eta}_{K,\sigma} + \mathcal{M}_K(\boldsymbol{U}_K)\boldsymbol{\varepsilon}_K^0$$

Thus, we deduce that, using Cauchy-Schwarz inequality

$$m_K \left|\nabla_K(\boldsymbol{U}_K)\right|^2 \leqslant \frac{2}{m_K} \left(\sum_{\sigma\in\mathcal{F}_K} m_K h_K^{-2}(u_\sigma - \mathcal{M}_K(\boldsymbol{U}_K))^2\right)\left(\sum_{\sigma\in\mathcal{F}_K} h_K^2 m_K^{-1} \left|\boldsymbol{\eta}_{K,\sigma}\right|^2\right) + 2m_K^2 |\mathcal{M}_K(\boldsymbol{U}_K)|^2 |\boldsymbol{\varepsilon}_K^0|^2$$

$$\leqslant 2\left(\sum_{\sigma\in\mathcal{F}_K} m_K h_K^{-2}(u_\sigma - \mathcal{M}_K(\boldsymbol{U}_K))^2\right)\left(\sum_{\sigma\in\mathcal{F}_K} h_K^2 \left|\frac{\boldsymbol{\eta}_{K,\sigma}}{m_K}\right|^2\right) + 2d m_K^2 \theta_{\mathcal{A}}^2 h_K^{2p} |\mathcal{M}_K(\boldsymbol{U}_K)|^2$$

from which the result immediately follows. □

We immediately deduce:

Lemma 2.4. *For any $(\boldsymbol{U},\boldsymbol{V}) \in X_{\mathcal{N},K}^2$*

$$a_h^K(\boldsymbol{U},\boldsymbol{V}) \leqslant \left((2\theta_\Pi^2\theta_{\mathcal{F}} + 2d\theta_{\mathcal{A}}^2 h^{2p}) + S^*(2 + 4\theta_\Pi^2\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^4 + 4d\theta_{\mathcal{A}}^2\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^3 h^{2p})\right) ||\boldsymbol{U}||_{X,K} ||\boldsymbol{V}||_{X,K} \tag{26}$$

while for any $(\boldsymbol{U},\boldsymbol{V}) \in X_{\mathcal{N}}^2$

$$a_h(\boldsymbol{U},\boldsymbol{V}) \leqslant \left((2\theta_\Pi^2\theta_{\mathcal{F}} + 2d\theta_{\mathcal{A}}^2 h^{2p}) + S^*(2 + 4\theta_\Pi^2\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^4 + 4d\theta_{\mathcal{A}}^2\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^3 h^{2p})\right) ||\boldsymbol{U}||_X ||\boldsymbol{V}||_X \tag{27}$$

For any $\boldsymbol{U} \in X_{\mathcal{N},K}$

$$a_h^K(\boldsymbol{U},\boldsymbol{U}) \geqslant \frac{S_*}{1+\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^3 S_*}|\boldsymbol{U}|_{1,K}^2 \tag{28}$$

while for any $\boldsymbol{U} \in X_{\mathcal{N}}$*:*

$$a_h(\boldsymbol{U},\boldsymbol{U}) \geqslant \frac{S_*}{1+\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^3 S_*}|\boldsymbol{U}|_1^2 \tag{29}$$

Finally, for any $\boldsymbol{V} \in X_{\mathcal{N}}$

$$l_h(\boldsymbol{V}) \leqslant C_f|\boldsymbol{V}|_0 \quad \textit{where} \quad C_f = \left(\sum_{K\in\mathcal{T}} m_K|f_K|^2\right)^{\frac{1}{2}} \tag{30}$$

Proof. Using Cauchy-Schwarz inequality, we immediately get

$$l_h(\boldsymbol{V}) \leqslant \left(\sum_{K\in\mathcal{T}} m_K|f_K|^2\right)^{\frac{1}{2}}\left(\sum_{K\in\mathcal{T}} m_K|\mathcal{M}_K(\boldsymbol{V}_K)|^2\right)^{\frac{1}{2}} \leqslant C_f|\boldsymbol{V}|_0$$

Next, still using Cauchy-Schwarz inequality, we get

$$\sum_{K\in\mathcal{T}} s^K(\boldsymbol{U}_K - \mathcal{D}_K\left(\Pi_K\left(\boldsymbol{U}_K\right)\right), \boldsymbol{V}_K - \mathcal{D}_K\left(\Pi_K\left(\boldsymbol{V}_K\right)\right))$$

$$\leqslant S^*\left(\sum_{K\in\mathcal{T}} m_K h_K^{-2}|\boldsymbol{U}_K - \mathcal{D}_K\left(\Pi_K\left(\boldsymbol{U}_K\right)\right)|^2\right)^{\frac{1}{2}}\left(\sum_{K\in\mathcal{T}} m_K h_K^{-2}|\boldsymbol{V}_K - \mathcal{D}_K\left(\Pi_K\left(\boldsymbol{V}_K\right)\right)|^2\right)^{\frac{1}{2}}$$

Notice that

$$\boldsymbol{U}_K - \mathcal{D}_K\left(\Pi_K(\boldsymbol{U}_K)\right) = \boldsymbol{U}_K - \mathcal{M}_K(\boldsymbol{U}_K) - \mathcal{D}_K\left(\nabla_K(\boldsymbol{U}_K)\cdot(\boldsymbol{x}-\overline{\boldsymbol{x}}_K)\right)$$

and that for any $(\sigma,\sigma') \in \mathcal{F}_K^2$, by construction we have $|\boldsymbol{x}_\sigma - \boldsymbol{x}_{\sigma'}| \leqslant h_K$, so we have as $\sum_{\sigma\in\mathcal{F}_K}\gamma_{K,\sigma} = 1$:

$$|\mathcal{D}_K\left(\nabla_K(\boldsymbol{U}_K)\cdot(\boldsymbol{x}-\overline{\boldsymbol{x}}_K)\right)| \leqslant \theta_{\mathcal{M}}\theta_{\mathcal{F}} h_K\,|\nabla_K(\boldsymbol{U}_K)|$$

which implies

$$|\boldsymbol{U}_K - \mathcal{D}_K\left(\Pi_K(\boldsymbol{U}_K)\right)|^2 \leqslant 2\sum_{\sigma\in\mathcal{F}_K}|u_\sigma - \mathcal{M}_K(\boldsymbol{U}_K)|^2 + 2h_K^2\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^3\,|\nabla_K(\boldsymbol{U}_K)|^2$$

Then, lemma 2.3 gives

$$\sum_{K\in\mathcal{T}}\sum_{\sigma\in\mathcal{F}_K} m_K h_K^{-2}|u_\sigma - \mathcal{M}_K(\boldsymbol{U}_K)|^2 + m_K\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^3\,|\nabla_K(\boldsymbol{U}_K)|^2 \leqslant (1+2\theta_\Pi^2\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^4)|\boldsymbol{U}|_1^2 + 2d\theta_{\mathcal{A}}\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^3 h^{2p}|\boldsymbol{U}|_0^2$$

and we deduce that

$$\sum_{K\in\mathcal{T}} s^K(\boldsymbol{U}_K - \mathcal{D}_K\left(\Pi_K\left(\boldsymbol{U}_K\right)\right), \boldsymbol{V}_K - \mathcal{D}_K\left(\Pi_K\left(\boldsymbol{V}_K\right)\right)) \leqslant 2S^*(1+2\theta_\Pi^2\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^4)|\boldsymbol{U}|_1|\boldsymbol{V}|_1 + 4dS^*\theta_{\mathcal{A}}\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^3 h^{2p}|\boldsymbol{U}|_0|\boldsymbol{V}|_0$$

Thus, using again Cauchy-Schwarz inequality and lemma 2.3 leads to

$$a_h(\boldsymbol{U},\boldsymbol{V}) \leqslant \left((2\theta_\Pi^2\theta_{\mathcal{F}} + 2d\theta_{\mathcal{A}}^2 h^{2p}) + S^*(2+4\theta_\Pi^2\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^4 + 4d\theta_{\mathcal{A}}^2\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^3 h^{2p})\right)||\boldsymbol{U}||_X||\boldsymbol{V}||_X$$

Finally, using the identity

$$(a-b)^2 \geqslant \frac{\rho}{1+\rho}a^2 - \rho\,b^2 \quad \forall\,(a,b)\in\mathbb{R}^2,\ \forall\,\rho > -1 \tag{31}$$

proceeding as above we see that, for all $\rho > -1$

$$|\boldsymbol{U}_K - \boldsymbol{\mathcal{D}}_K\left(\Pi_K(\boldsymbol{U}_K)\right)|^2 \geqslant \frac{\rho}{1+\rho}\sum_{\sigma\in\mathcal{F}_K}|u_\sigma - \mathcal{M}_K(\boldsymbol{U}_K)|^2 - \rho h_K^2\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^3\,|\nabla_K(\boldsymbol{U}_K)|^2$$

and thus:

$$a_h(\boldsymbol{U},\boldsymbol{U}) \geqslant \left(1-\rho\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^3 S_*\right)\sum_{K\in\mathcal{T}} m_K\,|\nabla_K(\boldsymbol{U}_K)|^2 + \frac{\rho S_*}{1+\rho}\,|\boldsymbol{U}|_1^2$$

Taking $\rho = \frac{1}{\theta_{\mathcal{M}}^2\theta_{\mathcal{F}}^3 S_*}$ gives the desired result. The proofs of the local versions are exactly the same, simply omitting the sums over $K \in \mathcal{T}$. □

Notice that as $X_{\mathcal{N},0}$ has finite dimension, uniqueness is sufficient to get existence, then the above results gives existence and uniqueness of the discrete solution. Indeed, all norms being equivalent in the finite dimensional setting, as $|.|_1$ is obviously a norm on $X_{\mathcal{N},0}$ estimate (29) is thus a coercivity estimate for this norm . This means that at this step we already know that it makes sense to consider the solution given by the network element method. However a discrete Poincaré's inequality is still missing if we wish to establish the stability of the method in the $||\cdot||_X$ normwith network independent constants. We recall the following definition: a domain Ω of $\mathbb{R}^d$ is said to satisfy the cone condition with angle τ and radius r if for any $\boldsymbol{x} \in \Omega$, there exists $\boldsymbol{\xi} \in \mathbb{R}^d$ with $|\xi| = 1$ such that $C(x,\boldsymbol{\xi},\tau,r) \subset \Omega$ where $C(x,\boldsymbol{\xi},\tau,r)$ denotes the cone

$$C(x,\boldsymbol{\xi},\tau,r) = B(\boldsymbol{x},r) \cap \left\{\boldsymbol{y}\in\mathbb{R}^d \mid (\boldsymbol{y}-\boldsymbol{x})^T\boldsymbol{\xi} > |\boldsymbol{y}-\boldsymbol{x}|cos\,\tau\right\} \tag{32}$$

and we recall the well known result that if Ω is Lipschitz then it satisfies the cone condition for some τ and r (see [27, 28]). Introducing the last measure of quality of the discretization network, we set

$$\theta_{\mathcal{T}} = \sup_{K\in\mathcal{T}}\max\left(\frac{|B_K\cap\Omega|}{m_K}, \frac{m_K}{|B_K\cap\Omega|}\right)$$

and we denote $S_1^d = |B(0,1)|$ the volume of the unit ball in dimension d. Then, we have

Lemma 2.5. *Assume that Ω satisfies the cone condition with angle τ and radius r, and denote $\delta > 0$ the smallest real number such that for any $K \in \mathcal{T}$*

$$\delta^{-1} r_K \leqslant min(r, r_K) \leqslant \delta r_K \tag{33}$$

Then there exists $C_{P,X_N} > 0$ depending on τ, δ, $\eta_{\mathcal{N}}$, $\theta_{\mathcal{T}}$, and Ω such that, for any $\boldsymbol{U} \in X_{\mathcal{N},0}$

$$||\boldsymbol{U}||_0^2 \leqslant C_{P,X_N}\,|\boldsymbol{U}|_1^2$$

Proof. This proof is an adaptation of the proof of the discrete Poincaré's inequality of [29] in the case of finite volume meshes.

Let $\boldsymbol{\nu} \in \mathbb{R}^d$ such that $||\boldsymbol{\nu}|| = 1$. For any $\boldsymbol{x} \in \Omega$, we denote $\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\nu}}$ the half-line with origin $\boldsymbol{x}$ and direction $\boldsymbol{\nu}$. Let $\boldsymbol{y}(\boldsymbol{x})$ be the point of $\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\nu}} \cap \partial\Omega$ such that $[\boldsymbol{x},\boldsymbol{y}(\boldsymbol{x})] \subset \overline{\Omega}$ and the length of $[\boldsymbol{x},\boldsymbol{y}(\boldsymbol{x})]$ is minimal (i.e. $\boldsymbol{y}(\boldsymbol{x})$ is the first point of $\partial\Omega$ along $\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\nu}}$).

Then, by construction of the network, there exists an integer $p < card(\mathcal{T})$, a family $(K_i)_{0\leqslant i\leqslant p}$ of elements of $\mathcal{T}$ and a family $(\sigma_{i,i+1})_{0\leqslant i\leqslant p-1}$ of elements of $\mathcal{F}$ such that for any $0 \leqslant i \leqslant p-1$, $\{K_i, K_{i+1}\} \subset \mathcal{T}_{\sigma_{i,i+1}}$, and such that $\boldsymbol{x} \in K_0$ and $\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\nu}}$ then goes through each of the $\mathcal{B}_{K_i}$'s once until reaching $\mathcal{B}_{K_p}$, with $\boldsymbol{y}(\boldsymbol{x}) \in \mathcal{B}_{K_p}$ and $\mathcal{F}_{ext} \cap \mathcal{F}_{K_p} \neq \varnothing$. Thus, we have

$$\mathcal{M}_K(\boldsymbol{U}) = \sum_{i=0}^{p-1}(\mathcal{M}_{K_i}(\boldsymbol{U}) - u_{\sigma_{i,i+1}}) + \sum_{i=0}^{p-1}(u_{\sigma_{i,i+1}} - \mathcal{M}_{K_{i+1}}(\boldsymbol{U})) + \mathcal{M}_{K_p}(\boldsymbol{U})$$

As there exists $\sigma_{p,p+1} \in \mathcal{F}_{K_p} \cap \mathcal{F}_{ext}$, for any $\boldsymbol{U} \in X_{\mathcal{N},0}$ as $u_{\sigma_{p,p+1}} = 0$, we have

$$\mathcal{M}_{K_p}(\boldsymbol{U}) = \mathcal{M}_{K_p}(\boldsymbol{U}) - u_{\sigma_{p,p+1}}$$

Denoting $\mathcal{T}_{\mathcal{L}_x^\nu}$ the family $(K_i)_{0\leqslant i\leqslant p}$, and $\mathcal{F}_{L,x}^\nu = \mathcal{F}_L \cap (\sigma_{i,i+1})_{0\leqslant i\leqslant p}$ for any $L \in \mathcal{T}$, it is consequently obvious that for a.e. $\boldsymbol{x} \in \mathcal{B}_K$

$$|\mathcal{M}_K(\boldsymbol{U})| \leqslant \sum_{L\in\mathcal{T}_{\mathcal{L}_x^\nu}} \sum_{\sigma\in\mathcal{F}_{L,x}^\nu} |u_\sigma - \mathcal{M}_L(\boldsymbol{U})|$$

Using the orientation defined by $\boldsymbol{\nu}$, we denote $\boldsymbol{x}_{L,0}^\nu$ the first intersection point of B_L with $\mathcal{L}_x^\nu$ and $\boldsymbol{x}_{L,1}^\nu$ the last one, and, we denote I_L^ν the subpart of $\mathcal{L}_x^\nu$ delimited by the intersection of $\mathcal{L}_x^\nu$ with B_L, i.e. the interval delimited by $\sigma_{L,-}^\nu$ and $\sigma_{L,+}^\nu$. As $\mathcal{L}_x^\nu$ goes through the B_{K_i}'s only once, then $\sigma_{i-1,i}$ and $\sigma_{i,i+1}$ are each either on the $\boldsymbol{x}_{L,0}^\nu$ or $\boldsymbol{x}_{L,1}^\nu$ side of $\mathcal{L}_x^\nu$, and each on opposite sides. In general, we will have $\sigma_{i-1,i}$ on the $\boldsymbol{x}_{L,0}^\nu$ side and $\sigma_{i,i+1}$ on the $\boldsymbol{x}_{L,1}^\nu$ side, but for very specific configurations on coarse networks the other case might nevertheless occur. We denote $\boldsymbol{x}_{L,\sigma}^\nu$ the point $\boldsymbol{x}_{L,k}^\nu$ corresponding to any $\sigma \in \mathcal{F}_{L,x}^\nu$.

Denoting $\mathcal{F}_x^\nu = \cup_{L\in\mathcal{T}_{\mathcal{L}_x^\nu}} \mathcal{F}_{L,x}^\nu$, for any $\sigma \in \mathcal{F}_x^\nu$, let $P_{L\sigma}$ be the projection of $B_L \cap \Omega$ onto the plane containing $\boldsymbol{x}_{L,\sigma}^\nu$ and orthogonal to $\boldsymbol{x}_L - \boldsymbol{x}_{L,\sigma}^\nu$. Defining $\chi_{L\sigma}$ on $\mathbb{R}^d \times \mathbb{R}^d$ by setting $\chi_{L\sigma}(\boldsymbol{x},\boldsymbol{y}) = 1$ if $P_{L\sigma} \cap [\boldsymbol{x},\boldsymbol{y}] \neq \varnothing$ and $\chi_{L\sigma}(\boldsymbol{x},\boldsymbol{y}) = 0$ otherwise, we get

$$|\mathcal{M}_K(\boldsymbol{U})| \leqslant \sum_{L\in\mathcal{T}_{\mathcal{L}_x^\nu}} \sum_{\sigma\in\mathcal{F}_{L,x}^\nu} |u_\sigma - \mathcal{M}_L(\boldsymbol{U})| \chi_{L\sigma}(\boldsymbol{x},\boldsymbol{y}(\boldsymbol{x}))$$

Denoting $c_{L\sigma} = \left|(\boldsymbol{x}_L - \boldsymbol{x}_{L,\sigma}^\nu)\cdot\boldsymbol{\nu}\right|$, as $\boldsymbol{\nu}$ is fixed it is clear that we can find for any $K \in \mathcal{T}$ a $x \in B_K$ such that $c_{L\sigma} \neq 0$ for all $L \in \mathcal{T}_{\mathcal{L}_x^\nu}$ and all $\sigma \in \mathcal{F}_{L,x}^\nu$, thus using Cauchy-Schwarz inequality we get

$$|\mathcal{M}_K(\boldsymbol{U})|^2 \leqslant \left(\sum_{L\in\mathcal{T}_{\mathcal{L}_x^\nu}} \sum_{\sigma\in\mathcal{F}_{L,x}^\nu} \frac{|u_\sigma - \mathcal{M}_L(\boldsymbol{U})|^2}{c_{L\sigma}} \chi_{L\sigma}(\boldsymbol{x},\boldsymbol{y}(\boldsymbol{x}))\right)\left(\sum_{L\in\mathcal{T}_{\mathcal{L}_x^\nu}} \sum_{\sigma\in\mathcal{F}_{L,x}^\nu} c_{L\sigma}\chi_{L\sigma}(\boldsymbol{x},\boldsymbol{y}(\boldsymbol{x}))\right)$$

However, we have

$$\sum_{L\in\mathcal{T}_{\mathcal{L}_x^\nu}} \sum_{\sigma\in\mathcal{F}_{L,x}^\nu} c_{L\sigma}\chi_{L\sigma}(\boldsymbol{x},\boldsymbol{y}(\boldsymbol{x})) \leqslant \sum_{L\in\mathcal{T}_{\mathcal{L}_x^\nu}} \sum_{\sigma\in\mathcal{F}_{L,x}^\nu} c_{L\sigma} = \sum_{L\in\mathcal{T}_{\mathcal{L}_x^\nu}} \sum_{\sigma\in\mathcal{F}_{L,x}^\nu} \left|(\boldsymbol{x}_L - \boldsymbol{x}_{L,\sigma}^\nu)\cdot\boldsymbol{\nu}\right|$$

By construction, we have (see figure 2)

$$\sum_{\sigma\in\mathcal{F}_{L,x}^\nu} \left|(\boldsymbol{x}_L - \boldsymbol{x}_{L,\sigma}^\nu)\cdot\boldsymbol{\nu}\right| \leqslant |I_L^\nu|$$

Next, observe that each part of $\mathcal{L}_x^\nu$ is covered by at most $\eta_{\mathcal{N}}$ by intervals I_L^ν when summing over L, i.e.

$$\sum_{L\in\mathcal{T}_{\mathcal{L}_x^\nu}} |I_L^\nu| \leqslant \eta_{\mathcal{N}} diam(\Omega)$$

Thus:

$$|\mathcal{M}_K(\boldsymbol{U})|^2 \leqslant \eta_{\mathcal{N}}\ diam(\Omega)\left(\sum_{L\in\mathcal{T}} \sum_{\sigma\in\mathcal{F}_L} \frac{|u_\sigma - \mathcal{M}_L(\boldsymbol{U})|^2}{c_{L\sigma}} \chi_{L\sigma}(\boldsymbol{x},\boldsymbol{y}(\boldsymbol{x}))\right)$$

Then, multiplying by m_K and summing over $K \in \mathcal{T}$ leads to

$$\sum_{K\in\mathcal{T}} m_K|\mathcal{M}_K(\boldsymbol{U})|^2 = \sum_{K\in\mathcal{T}} \frac{m_K}{|B_K\cap\Omega|}\int_{B_K\cap\Omega} |\mathcal{M}_K(\boldsymbol{U})|^2$$

$$\leqslant \eta_{\mathcal{N}}\theta_{\mathcal{T}}\ diam(\Omega)\left(\sum_{L\in\mathcal{T}} \sum_{\sigma\in\mathcal{F}_L} \frac{|u_\sigma - \mathcal{M}_L(\boldsymbol{U})|^2}{c_{L\sigma}} \int_\Omega \sum_{K\in\mathcal{T}} \chi_{B_K}\chi_{L\sigma}(\boldsymbol{x},\boldsymbol{y}(\boldsymbol{x}))\right)$$

$$\leqslant \eta_{\mathcal{N}}^2\theta_{\mathcal{T}}\ diam(\Omega)\left(\sum_{L\in\mathcal{T}} \sum_{\sigma\in\mathcal{F}_L} \frac{|u_\sigma - \mathcal{M}_L(\boldsymbol{U})|^2}{c_{L\sigma}} \int_\Omega \chi_{L\sigma}(\boldsymbol{x},\boldsymbol{y}(\boldsymbol{x}))\right)$$

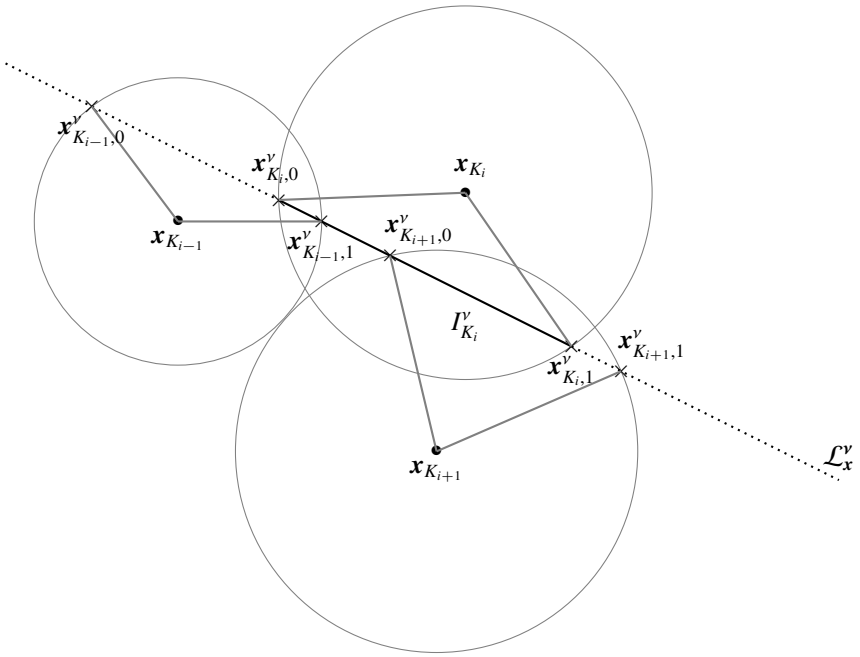

Figure 2: Illustration in dimension 2 for the bound on $\sum_{L\in\mathcal{T}_{\mathcal{L}^{\nu}_x}}\sum_{\sigma\in\mathcal{F}^{\nu}_{L,x}}\left|(\boldsymbol{x}_L-\boldsymbol{x}^{\nu}_{L,\sigma})\cdot\boldsymbol{\nu}\right|$

The set of $\boldsymbol{x}\in\Omega$ such that $\chi_{L\sigma}(\boldsymbol{x},\boldsymbol{y}(\boldsymbol{x}))\neq 0$ corresponds to

$$\Omega\cap\left\{\boldsymbol{x}\in\mathbb{R}^d \mid \boldsymbol{x}=\boldsymbol{y}_{L\sigma}-\xi\boldsymbol{\nu},\ \text{with}\ \boldsymbol{y}_{L\sigma}\in P_{L\sigma}\ \text{and}\ \xi>0\right\}$$

Thus, we have

$$\int_\Omega \chi_{L\sigma}(\boldsymbol{x},\boldsymbol{y}(\boldsymbol{x}))\leqslant\frac{|P_{L\sigma}|c_{L\sigma}}{||\boldsymbol{x}_L-\boldsymbol{x}^{\nu}_{L,\sigma}||}\,diam(\Omega)$$

and consequently

$$\sum_{K\in\mathcal{T}} m_K|\mathcal{M}_K(\boldsymbol{U})|^2\leqslant\eta_{\mathcal{N}}^2\theta_{\mathcal{T}}\,diam(\Omega)^2\left(\sum_{L\in\mathcal{T}}\sum_{\sigma\in\mathcal{F}_L}|u_\sigma-\mathcal{M}_L(\boldsymbol{U})|^2\frac{|P_{L\sigma}|}{||\boldsymbol{x}_L-\boldsymbol{x}^{\nu}_{L,\sigma}||}\right)$$

$$\leqslant\eta_{\mathcal{N}}^2\theta_{\mathcal{T}}\,diam(\Omega)^2\left(\sum_{L\in\mathcal{T}}\sum_{\sigma\in\mathcal{F}_L}|u_\sigma-\mathcal{M}_L(\boldsymbol{U})|^2\frac{|P_{L\sigma}|}{r_L}\right)$$

as $||\boldsymbol{x}_L-\boldsymbol{x}^{\nu}_{L,\sigma}||=r_L$ by construction. Notice that $|P_{L\sigma}|\leqslant S_1^{d-1}r_L^{d-1}$. As Ω satisfies the cone condition, notice then that for any $L\in\mathcal{T}$, there exists $\boldsymbol{\xi}\in\mathbb{R}^d$ such that

$$|B_L\cap\Omega|\geqslant|C(\boldsymbol{x}_L,\boldsymbol{\xi},\tau,min(r,r_L))|=|C(0,\boldsymbol{\xi},\tau,1)|min(r,r_L)^d$$

Noticing that as $|C(0,\boldsymbol{\xi},\tau,1)|$ is in fact independent on $\boldsymbol{\xi}$, we can denote$|C(0,\boldsymbol{\xi},\tau,1)|=|C(0,\tau,1)|$ this common value and get

$$|B_L\cap\Omega|\geqslant|C(0,\tau,1)|min(r,r_L)^d$$

Then, we get

$$|P_{L\sigma}|=m_L\frac{|P_{L\sigma}|}{m_L}=m_L\frac{|B_L\cap\Omega|}{m_L}\frac{|P_{L\sigma}|}{|B_L\cap\Omega|}\leqslant m_L\theta_{\mathcal{T}}\frac{S_1^{d-1}r_L^{d-1}}{|C(0,\tau,1)|min(r,r_L)^d}\leqslant 2m_Lh_L^{-1}\theta_{\mathcal{T}}\delta^d\frac{S_1^{d-1}}{|C(0,\tau,1)|}$$

and thus

$$\sum_{K\in\mathcal{T}} m_K|\mathcal{M}_K(\boldsymbol{U})|^2 \leqslant 4\eta_{\mathcal{N}}^2\ diam(\Omega)^2\theta_{\mathcal{T}}^2\delta^d\frac{S_1^{d-1}}{|C(0,\tau,1)|}\left(\sum_{L\in\mathcal{T}}\sum_{\sigma\in\mathcal{F}_L} m_L h_L^{-2}|u_\sigma - \mathcal{M}_L(\boldsymbol{U})|^2\right)$$

and the result follows with

$$C_{P,X_{\mathcal{N}}} = 4\eta_{\mathcal{N}}^2\ diam(\Omega)^2\theta_{\mathcal{T}}^2\delta^d\frac{S_1^{d-1}}{|C(0,\tau,1)|}$$

□

Remark 2.6. Immediately, we notice that when h goes to zero, as r is a fixed property of Ω we can take $\delta = 1$, implying that the dependency in δ of the discrete Poincaré's constant will never be a major issue.

This discrete Poincaré's inequality immediately implies that the norm $|\cdot|_1$ on $X_{\mathcal{N},0}$ is equivalent to the $||\cdot||_X$ on $X_{\mathcal{N},0}$, which now makes existence, uniqueness and stability of the discrete solution an obvious consequence of Lax-Milgram's lemma

Proposition 2.7. *Assume that Ω satisfies the cone condition with angle τ and radius r. Let $\mathcal{N}$ be a discretization network and $\mathcal{G}$ an associated admissible network geometry. Then there exists a unique solution to the discrete problem*

$$\left|\begin{array}{l} \text{Find } \boldsymbol{U}_h \in X_{\mathcal{N},0} \text{ such that} \\ a_h(\boldsymbol{U}_h, \boldsymbol{V}_h) = l_h(\boldsymbol{V}_h) \text{ for all } \boldsymbol{V}_h \in X_{\mathcal{N},0} \end{array}\right.$$

and a constant $C > 0$ depending on τ, δ, $\eta_{\mathcal{N}}$, $\theta_{\mathcal{F}}$, $\theta_{\mathcal{T}}$, $\theta_{\mathcal{A}}$ and Ω but not on h such that:

$$||\boldsymbol{U}_h||_X \leqslant C||\boldsymbol{F}||_0$$

where δ is defined by (33) *and $\boldsymbol{F} = (f_K)_{K\in\mathcal{T}}$.*

Let us conclude this preliminary analysis by some comments on the quality measures of the discretization network and its geometry. While $\theta_{\mathcal{F}}$ is clearly a bound on its connectivity, as one would naturally require on a mesh. Next, a closer look to θ_Π reveals that it behaves like a local measure of the network deformation and combined with $\theta_{\mathcal{M}}$ and $\theta_{\mathcal{F}}$ it corresponds to the usual control over shape regularity or, from a classical finite element perspective, one could say that it corresponds to the control over the number of shapes for reference elements. At first the quantity $\theta_{\mathcal{T}}$ seems to have no direct equivalent for classical meshes, as it quantifies in some sense the coherence between the covering diameter and the prescribed local measure m_K. However, as we will explain in the following section, it can bounded by an equivalent of the usual chunkiness parameter, and can thus be considered as its equivalent. However, $\theta_{\mathcal{A}}$ measuring the quality of geometric consistency and conservation approximations, it cannot have an equivalent for true meshes.

3. Numerical exploration

3.1. Practical construction of the network geometry

Assume that we are given an admissible discretization network $\mathcal{N}$. Ideally, together with the generation of the network the geometry generation step should remain competitive with the usual meshing and geometrical computation steps. Thus, in the worst case it should not require more computational time than a meshing algorithm (of course, we have in mind relatively complex algorithms like Delaunay or Voronoï mesh generation, or complex 3d mesh generation, as we cannot hope to be competitive with very simple and instantaneous algorithms such as pure Cartesian mesh generation without any kind of cut-cells). As our intention is to focus the present paper on the network element method itself rather than on the network and geometry generation, we will consider a quite generic approach, which is nevertheless far from being optimal in this regard. Possible paths for improving the performance of the geometry generation procedure will be briefly discussed at the end of this section.

To construct an admissible geometry, the most general procedure is to solve an optimization problem of the form

$$\mathcal{G} = \arg\min_{\hat{\mathcal{G}} \in \mathcal{A}_{\mathcal{G}}} \mathcal{J}(\hat{\mathcal{G}}) \tag{34}$$

where $\mathcal{A}_{\mathcal{G}}$ is the set of admissible geometries, i.e. satisfying (6)-(4)-(5)-(7). Obviously, to reduce the computational effort we will not require a global minimum to our minimization problem. Moreover the cost function $\mathcal{J}$ must remain relatively simple, so we can control the computational cost of those minimization problems to a large extent. Strictly speaking any cost function $\mathcal{J}(\hat{\mathcal{G}})$ could be used, however it seems relevant to choose one that allows some control over the quality of the generated geometry i.e. on the geometry dependent constants appearing in our stability and error estimates, while keeping a correct trade-off between the optimality of those constants and the computational cost of the above minimization problems. In the present paper we restrict ourselves to the simplest cost function given by

$$\begin{aligned}\mathcal{J}(\hat{\mathcal{G}}) = \frac{\delta^m_{\mathcal{J}}}{2} \sum_{K\in\mathcal{T}} \hat{m}_K^2 + \frac{\delta^\eta_{\mathcal{J}}}{2} \sum_{K\in\mathcal{T}} \sum_{\sigma\in\mathcal{F}_K} |\hat{\boldsymbol{\eta}}_{K,\sigma}|^2 + \frac{1}{2} \sum_{K\in\mathcal{T}} \omega^0_{\mathcal{J},K} |\hat{\boldsymbol{\varepsilon}}^0_{m,K}|^2 + \frac{1}{2} \sum_{K\in\mathcal{T}} \sum_{i=1}^d \sum_{j=1}^d \omega^1_{\mathcal{J},K} \left|\hat{\varepsilon}^{1,ij}_{m,K}\right|^2 \\ + \frac{1}{2} \sum_{\sigma\in\mathcal{F}_{int}} \omega_{\mathcal{J},\sigma} |\hat{\boldsymbol{\varepsilon}}_\sigma|^2 + \frac{1}{2} \omega_{\mathcal{J},\Omega} |\hat{\varepsilon}_\Omega|^2\end{aligned} \tag{35}$$

where $\delta^m_{\mathcal{J}}$ and $\delta^\eta_{\mathcal{J}}$ are penalization parameters and where we have used the change of variable $\hat{\boldsymbol{\varepsilon}}^0_{m,K} = m_K \hat{\boldsymbol{\varepsilon}}^0_K$ and $\hat{\varepsilon}^{1,ij}_{m,K} = m_K \hat{\varepsilon}^{1,ij}_K$. This change of variable leads to a quadratic optimization problem with linear equality and inequality constraints, and is very similar to the one of [22].

Let us explain how to choose the weights involved in formula (35). The Lagrangian associated with the minimization problem (34) is given by

$$\begin{aligned}\mathcal{L}(\hat{\mathcal{G}}, \hat{\lambda}) = \mathcal{J}(\hat{\mathcal{G}}) + \sum_{K\in\mathcal{T}} \sum_{i=1}^d \left(\sum_{\sigma\in\mathcal{F}_K} \hat{\eta}^i_{K,\sigma} - \hat{\varepsilon}^{0,i}_K \right) \hat{\lambda}^{0,i}_K + \sum_{K\in\mathcal{T}} \sum_{i=1}^d \sum_{j=1}^d \left(\hat{\eta}^i_{K,\sigma} (x^j_\sigma - x^j_K) - \hat{m}_K \delta_{ij} - \hat{\varepsilon}^{1,ij}_K \right) \hat{\lambda}^{1,ij}_K \\ + \sum_{\sigma\in\mathcal{F}_{int}} \left(\sum_{K\in\mathcal{T}_\sigma} \hat{\eta}^i_{K,\sigma} - \hat{\varepsilon}^i_\sigma \right) \hat{\lambda}^i_\sigma + \left(\sum_{K\in\mathcal{T}} \hat{m}_K - |\Omega| - \hat{\varepsilon}_\Omega \right) \hat{\lambda}_\Omega + \sum_{K\in\mathcal{T}} \hat{m}_K (\hat{\lambda}^{max}_K - \hat{\lambda}^{min}_K)\end{aligned}$$

where:

$$\hat{\lambda} = \left((\hat{\lambda}_K)_{K\in\mathcal{T}}, (\hat{\lambda}^{0,i}_K)_{K\in\mathcal{T}, 1\leqslant i\leqslant d}, (\hat{\lambda}^{1,ij}_K)_{K\in\mathcal{T}, 1\leqslant i,j\leqslant d}, (\hat{\lambda}^i_\sigma)_{\sigma\in\mathcal{F}_{int}, 1\leqslant i\leqslant d}, \hat{\lambda}_\Omega \right)$$

denotes the set of Lagrange multipliers. The Karush-Kuhn-Tucker first order optimality conditions associated with the system are given by (setting $\hat{\lambda}^i_\sigma = 0$ for any $\sigma \in \mathcal{F}_{ext}$ and any $1 \leqslant i \leqslant d$)

$$\partial_{\hat{\varepsilon}^i_\sigma} \mathcal{L}(\hat{\mathcal{G}}, \hat{\lambda}) = \omega_{\mathcal{J},\sigma} \hat{\varepsilon}^i_\sigma - \hat{\lambda}^i_\sigma = 0 \tag{36}$$

$$\partial_{\hat{\varepsilon}^{0,i}_K} \mathcal{L}(\hat{\mathcal{G}}, \hat{\lambda}) = \omega^0_{\mathcal{J},K} \hat{\varepsilon}^{0,i}_K - \hat{\lambda}^{0,i}_K = 0 \tag{37}$$

$$\partial_{\hat{\varepsilon}^{1,ij}_K} \mathcal{L}(\hat{\mathcal{G}}, \hat{\lambda}) = \omega^1_{\mathcal{J},K} \hat{\varepsilon}^{1,ij}_K - \hat{\lambda}^{1,ij}_K = 0 \tag{38}$$

$$\partial_{\hat{\eta}^i_{K,\sigma}} \mathcal{L}(\hat{\mathcal{G}}, \hat{\lambda}) = \delta^\eta_{\mathcal{J}} \hat{\eta}^i_{K,\sigma} + \sum_{j=1}^d \hat{\lambda}^{1,ij}_K (x^j_\sigma - x^j_K) + \hat{\lambda}^{0,i}_K + \hat{\lambda}^i_\sigma = 0 \tag{39}$$

$$\partial_{\hat{m}_K} \mathcal{L}(\hat{\mathcal{G}}, \hat{\lambda}) = \delta^m_{\mathcal{J}} \hat{m}_K - \sum_{i=1}^d \hat{\lambda}^{1,ii}_K + \hat{\lambda}^{max}_K - \hat{\lambda}^{min}_K + \hat{\lambda}_\Omega = 0 \tag{40}$$

$$\partial_{\hat{\varepsilon}_\Omega} \mathcal{L}(\hat{\mathcal{G}}, \hat{\lambda}) = \omega_{\mathcal{J},\Omega} \hat{\varepsilon}_\Omega - \hat{\lambda}_\Omega = 0 \tag{41}$$

complemented by the constraints (4)-(5), (6)-(7) and the complementarity conditions

$$\Upsilon_{comp}(\hat{m}_K, \hat{\lambda}^{min}_K) = 0 \quad \text{and} \quad \Upsilon_{comp}(\tau|B_K| - \hat{m}_K, \hat{\lambda}^{max}_K) = 0$$

where Υ_{comp} can be any complementarity function, such the minimum function or Fischer-Burmeister function, and $\tau > 1$ a fixed parameter. The upper bound constraint on the measures is introduced for reasons that will immediately become clear. Indeed, using our analogy between network geometries and mesh geometries, let us denote

$$\hat{h}_K = \frac{1}{2} \max_{\sigma \in \mathcal{F}_K} |\boldsymbol{x}_\sigma - \boldsymbol{x}_K|$$

and assume that $\hat{m}_K$ behaves as $\hat{h}_K^d$ and $\eta_{K,\sigma}^i$ as $\hat{h}_K^{d-1}$ and the penalization parameters as constants independent on $\hat{h}_K$. The upper bound constraint on the measures is a practical way to ensure that it will effectively be the case. Then, in the worst case, we have that

$$\hat{\lambda}_K^{1,ij} \sim \hat{h}_K^{d-2} \quad \text{and} \quad \hat{\lambda}_K^{0,i} \sim \hat{h}_K^{d-1} \quad \text{and} \quad \hat{\lambda}_\sigma^i \sim \hat{h}_K^{d-1} \quad \text{and} \quad \hat{\lambda}_\Omega \sim \hat{h}_K^{d-2}$$

Thus, to enforce that the approximation errors will be related to the local radii, in other words that h_K will behave like $\hat{h}_K$, it seems relevant to choose

$$\omega_{\mathcal{J},K}^0 = \frac{1}{\hat{h}_K^{p+1}} \qquad \omega_{\mathcal{J},K}^1 = \frac{1}{\hat{h}_K^{p+2}} \qquad \omega_{\mathcal{J},\sigma} = \frac{1}{\min_{K \in \mathcal{T}_\sigma} \hat{h}_K^{p+1}} \qquad \omega_{\mathcal{J},\Omega} = \frac{1}{\min_{K \in \mathcal{T}} \hat{h}_K^{p+2}}$$

Numerical experiments confirm that the above choice indeed gives the expected behavior, the practical choice for the remaining weights being $\delta_{\mathcal{J}}^m = 0.1$ and $\delta_{\mathcal{J}}^\eta = 0.01$. The minimization problem (34) with cost function (35) is solved in practice by applying Newton-Raphson's algorithm to the corresponding system of Karush-Kuhn-Tucker first order optimality conditions.

Even if our intention in the present paper is not to describe the most efficient way to compute a network geometry, let us make some comments on the way we solve this optimization problem in practice. As noticed in [22], efficiently solving this problem requires to take its specific structure into account. Indeed, it is extremely important to notice that once linearized, most unknowns can be eliminated through static condensation (i.e. Schur's complement), reducing the linear system to the face Lagrange multipliers $\hat{\lambda}_\sigma^i$ and the global Lagrange multiplier $\hat{\lambda}_\Omega$, i.e. a system of size $1 + d\,card(\mathcal{F}_{int})$. Several linear solvers have been tested to invert the involved linear systems, the best choices for our set of experiments seem to be GMRES and LSQR as they can handle efficiently linear systems with a non empty kernel or close to ones possessing a non empty kernel. Finally, let us mention that a good initial point for the quadratic optimization is obtained by solving the same system without the conservation constraint (7). In this case, all the cell unknowns can be eliminated by static condensation in the resulting linear system, leaving only $\hat{\lambda}_\Omega$ as a global variable. This initialization routine thus only involves small matrix inversion in each cell to perform static condensation, and in general a single Newton iteration of the overall optimization procedure.

As the minimization problem can only be approximately solved in practice, if $\mathcal{G}^*$ denotes the approximate solution of the minimization problem at the end of the process we need to recompute the exact errors $\varepsilon(\mathcal{G}^*)$ directly from (4)-(5), (6)-(7) applied to $\mathcal{G}^*$. In other words, we set

$$\overline{\varepsilon}_K^{0,i} = \frac{1}{m_K^*} \sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^{i,*} \quad \forall K \in \mathcal{T},\ \forall\, 1 \leqslant i \leqslant d$$

and

$$\overline{\varepsilon}_K^{1,ij} = \frac{1}{m_K^*} \sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^{i,*} (x_\sigma^j - x_K^j) - \delta_{ij} \quad \forall K \in \mathcal{T},\ \forall\, 1 \leqslant i, j \leqslant d$$

and

$$\overline{\varepsilon}_\Omega = \sum_{K \in \mathcal{T}} m_K^* - |\Omega| \quad \text{and} \quad \overline{\varepsilon}_\sigma^i = \sum_{K \in \mathcal{T}_\sigma} \eta_{K\sigma}^{i,*} \quad \forall\, \sigma \in \mathcal{F}_{int},\ \forall\, 1 \leqslant i \leqslant d$$

This allows to take into account the solver error accurately. Finally, let us mention that if $\theta_{\mathcal{A}}$ and θ_Π are directly computable from the network's geometry, parameter $\theta_{\mathcal{T}}$ is more delicate to obtain. However, it can easily be bounded by considering for any $K \in \mathcal{T}$ the inradius

$$\hat{\rho}_K = \min_{\sigma \in \mathcal{F}_K} |\boldsymbol{x}_\sigma - \boldsymbol{x}_K|$$

and then computing:

$$\theta_{\mathcal{T}}^{sup} = \max_{K\in\mathcal{T}} \max \left(\frac{m_K}{|B(\boldsymbol{x}_K, \hat{\rho}_K)|}, \frac{|B_K|}{m_K} \right)$$

For polygonal domains provided all points defining the boundary of Ω are included in the set of interface points, we clearly have $\theta_{\mathcal{T}} \leqslant \theta_{\mathcal{T}}^{sup}$. For domains with curved boundaries, the above $\theta_{\mathcal{T}}^{sup}$ will only be an approximation of an upper bound for $\theta_{\mathcal{T}}$ if the discretization is too coarse, that will become more and more precise as we refine the discretization network. Nevertheless, as it is anyway a very strong upper bound even for polygonal domains, the true $\theta_{\mathcal{T}}$ being in general much smaller, in the following we will mostly use the easily computable $\theta_{\mathcal{T}}^{sup}$ to assess the behavior of $\theta_{\mathcal{T}}$ for the studied geometries.

3.2. Mesh based networks

To conduct a first numerical study of the network element method (NEM), we start by considering some classical numerical test cases and the associated meshes. Although it can seem surprising to consider meshes as we have done all that we could to derive a completely meshless method, the interest of doing so is two fold: first, it will allow an easy comparison with the first order virtual element method (VEM) as we can use VEMs on the underlying meshes, and also using classical difficult and distorted meshes, we will have a first glance at the robustness of the NEM on distorted networks. By convention, we will set $\theta_{\mathcal{A}} = 1$ for true mesh geometries.

Let us mention that although in the present paper we build discretization networks out of meshes mainly for comparison purposes, such networks can be of practical interest. Indeed, it is not uncommon to encounter in practice workflows that start from badly shaped meshes only intended for data representation and that nevertheless keep those meshes for simulation. The reason for doing so is that generating a new mesh that has a sufficient numerical quality while retaining the main features from such data-oriented meshes can prove to be very difficult. Resorting to the network element method could be a good alternative in such situations, as it is not directly constrained by mesh cells quality.

We consider two ways of generating a network and its associated geometry starting from a mesh. Both will use the cell barycenters and diameters to define the point cloud and its connectivity. The first way of getting a network and its associated approximate geometry is very simple: it consists in choosing the interfaces to be the faces of the mesh, and to define the approximate geometry by perturbing the exact mesh geometry, i.e.

$$m_K = |K| + \frac{\theta_{\mathcal{A}}^{rand}|K|}{|\Omega|} diam(K)^p \omega_K \quad \forall\ K \in \mathcal{T}$$

and

$$\eta_{K,\sigma}^i = |\sigma| n_{K,\sigma}^i + \theta_{\mathcal{A}}^{rand} diam(K)^p \omega_{K,\sigma}^i \quad \forall\ K \in \mathcal{T},\ \forall\ \sigma \in \mathcal{F}_K,\ \forall\ 1 \leqslant i \leqslant d$$

where the ω_K's and $\omega_{K,\sigma}^i$'s are randomly generated numbers taking values in$]-1,1[$, and $\theta_{\mathcal{A}}^{rand}$ a fixed amplitude parameter, set to $\theta_{\mathcal{A}}^{rand} = 1$ in the following experiments. This will allow to study the impact on convergence of the approximation errors in a very simple way. This perturbed geometry approach will be denoted *NEM-GPG* (for "grid perturbed geometry") in the following tests.

The second way consists in constructing the discretization network by choosing the interfaces to be either the vertices of the mesh or some random perturbations of the vertices and then of course solving the minimization problem (34) with cost function (35). Those two approaches, vertex based and perturbed vertex based, will be denoted *NEM-GV* and *NEM-GPV* (for "grid vertices" and "grid perturbed vertices") in the following tests.

Finally, let us mention that to study the L^2 convergence of solutions, we will consider

$$\left(\sum_{K\in\mathcal{T}} m_K |\overline{u}(\overline{\boldsymbol{x}}_K) - \mathcal{M}_K(\boldsymbol{U})|^2 \right)^{1/2}$$

which, if the exact solution $\overline{u}$ is regular enough as in our tests cases, will be in principle a h^2 approximation of the true L^2 error. Notice that all the numerical tests in the following where performed with $p = 2$ and $\overline{\boldsymbol{x}}_K = \boldsymbol{x}_K$.

To assess the behavior of the method when the network is generated from a distorted mesh of $\Omega =]0,1[\times]0,1[$, we consider the test case

$$u(x,y) = sin(\pi x)sin(\pi y) \tag{42}$$

which will be named *Sinusoidal2D*. We consider four types of mesh sequences, from which we generate the discretization networks. The first one (*2dDualDelaunay*) is obtained by considering the dual meshes (the cell formed by joining the centers of the cell of the primal mesh) of a sequence of Delaunay meshes, while the second one (*2dVoronoi*) is obtained by considering the Voronoï meshes associated to the same sequence of Delaunay meshes. These two approaches generate polygonal meshes with quite generic cells. The third sequence (*2dKershawBox*) of meshes is a sequence of Kershaw meshes of the unit square, while the fourth one (*2dCheckerBoardBox*) is a sequence of checkerboard meshes of the unit square. These two sequences have only quadrangular cells which are distorted for the sequence *2dKershawBox* and non-conforming for the sequence *2dCheckerBoardBox*.

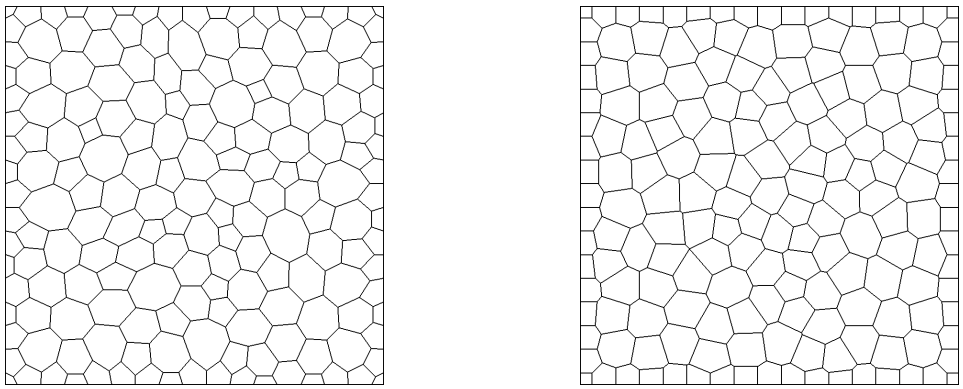

Figure 3: Example of meshes for the *2dDualDelaunay* and *2dVoronoi* mesh sequences

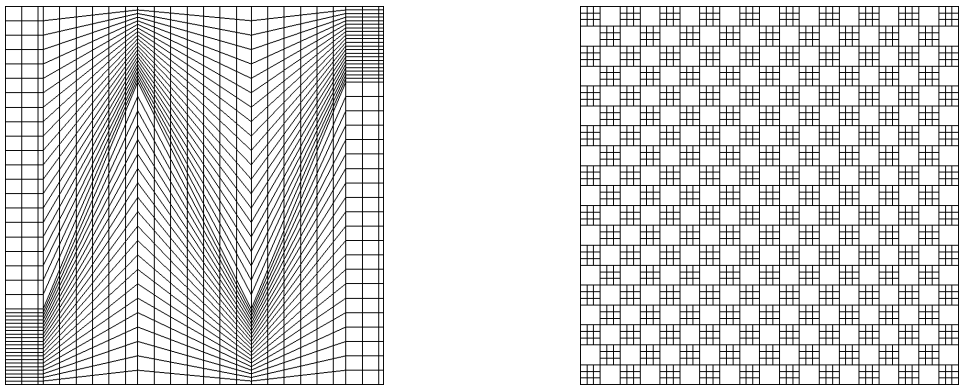

Figure 4: Example of meshes for the *2dKershawBox* and *2dCheckerBoardBox* mesh sequences

We display the convergence curves for each mesh sequence on figures 5 and 6, as well as the approximate convergence rates in table 1. We see that we recover the expected L^2 superconvergence, i.e. a convergence at rate h^2. Moreover, the different network geometries lead to schemes that have roughly the same behavior than the first order virtual element method (VEM) solved on the original mesh. Thus, the method is a reasonable alternative for those first tests cases. On figures 7-8, 9-10 and 11-12, we display the quality parameters $\theta_{\mathcal{T}}^{sup}$, θ_{Π} and $\theta_{\mathcal{A}}$ corresponding to the generated networks and also to the mesh itself.

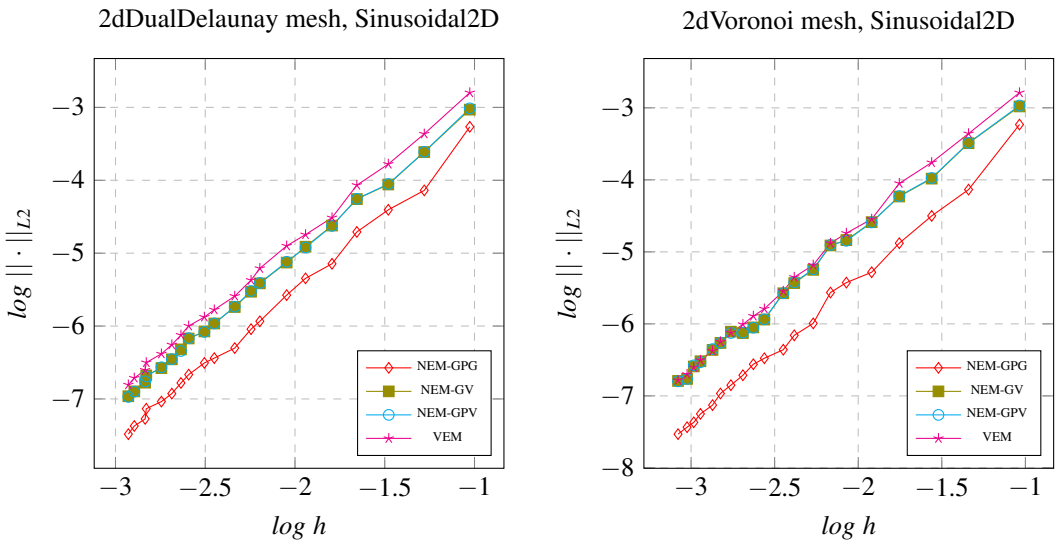

Figure 5: Convergence curves for *Sinusoidal2D* for the *2dDualDelaunay* and *2dVoronoi* mesh sequences

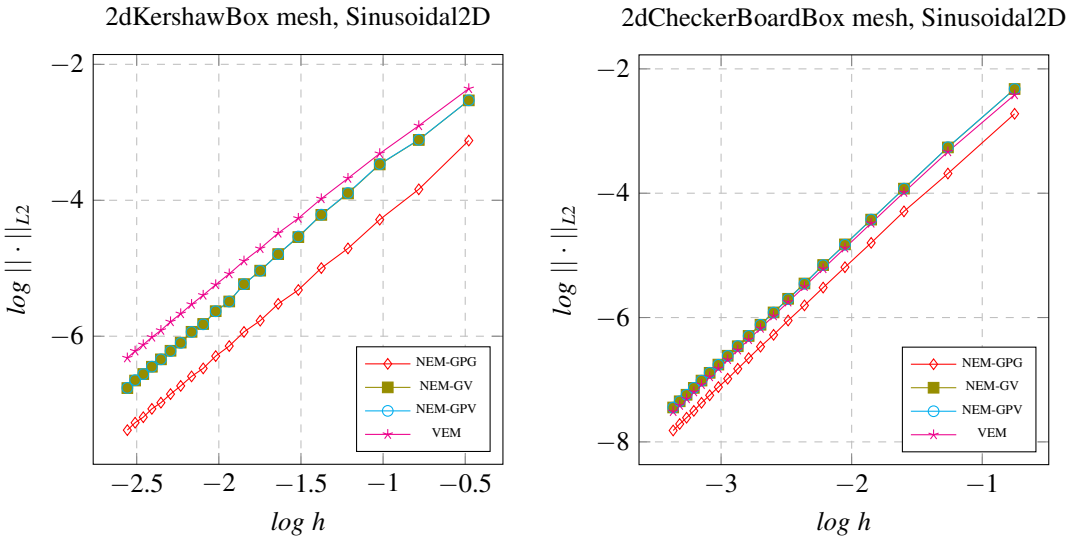

Figure 6: Convergence curves for *Sinusoidal2D* on the *2dKershawBox* and *2dCheckerBoardBox* mesh sequences

Table 1: Approximate orders of convergence for *Sinusoidal2D*

	2dDualDelaunay	*2dVoronoi*	*2dKershawBox*	*2dCheckerBoardBox*
VEM	2.065	2.024	1.973	1.999
NEM perturbed geometry	2.077	2.000	1.997	1.993
NEM vertex based	2.077	1.946	2.137	1.983
NEM perturbed vertex based	2.081	1.948	2.140	1.990

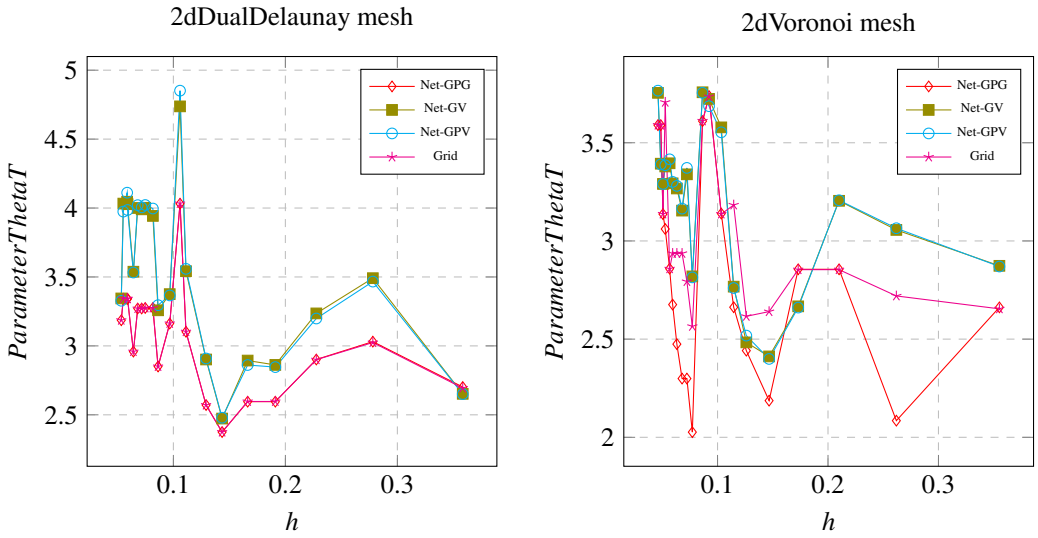

Figure 7: Quality parameter $\theta^{sup}_{\mathcal{T}}$ for the *2dDualDelaunay* and *2dVoronoi* mesh sequences

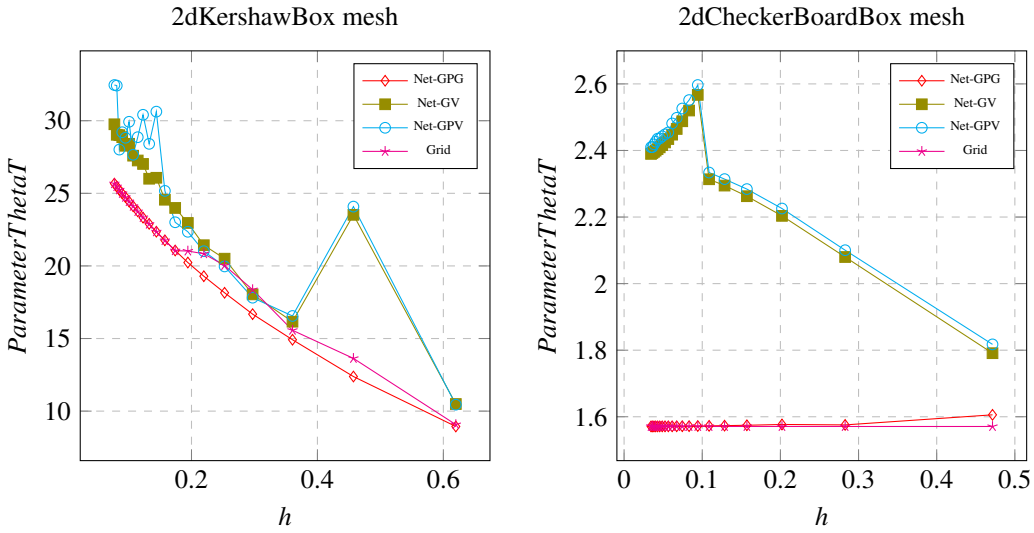

Figure 8: Quality parameter $\theta^{sup}_{\mathcal{T}}$ for the *2dKershawBox* and *2dCheckerBoardBox* mesh sequences

The first remark is that we indeed control those parameters for all the considered mesh sequences, as they do not blow up when h goes to zero. Next, we see that roughly speaking they more or less follow the behavior of their equivalent for the underlying mesh, revealing in particular that the network geometry generation procedure through optimization do not lead to unreasonable quality. Moreover, we see that the three sequences *2dDualDelaunay*, *2dVoronoi* and *2dCheckerBoardBox* are indeed perfectly regular sequences regarding those parameters. The situation is different for sequence *2dKershawBox* as expected: even if they remain under control, the three quality parameters $\theta^{sup}_{\mathcal{T}}$, θ_{Π} and $\theta_{\mathcal{A}}$ noticeably increase when we refine h for the Kershaw meshes. Interestingly, it is also the case for the mesh itself. Thus this slight degeneracy does not come from the fact that we have generated a network geometry, but rather from the spatial distribution of cells and interface, reflecting Kershaw meshes distortion. In fact, the Kershaw mesh sequence has been precisely devised for this purpose: it is not a completely regular mesh sequence, the usual mesh parameters such as the chunkiness parameter do degenerate when h diminishes. All in all, those first results validate both the

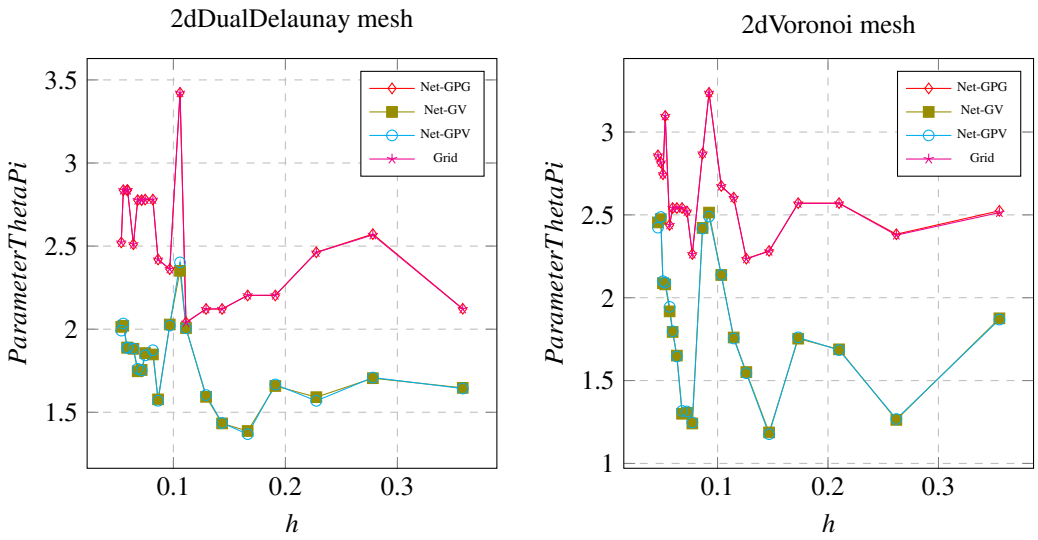

Figure 9: Quality parameter θ_Π for the *2dDualDelaunay* and *2dVoronoi* mesh sequences

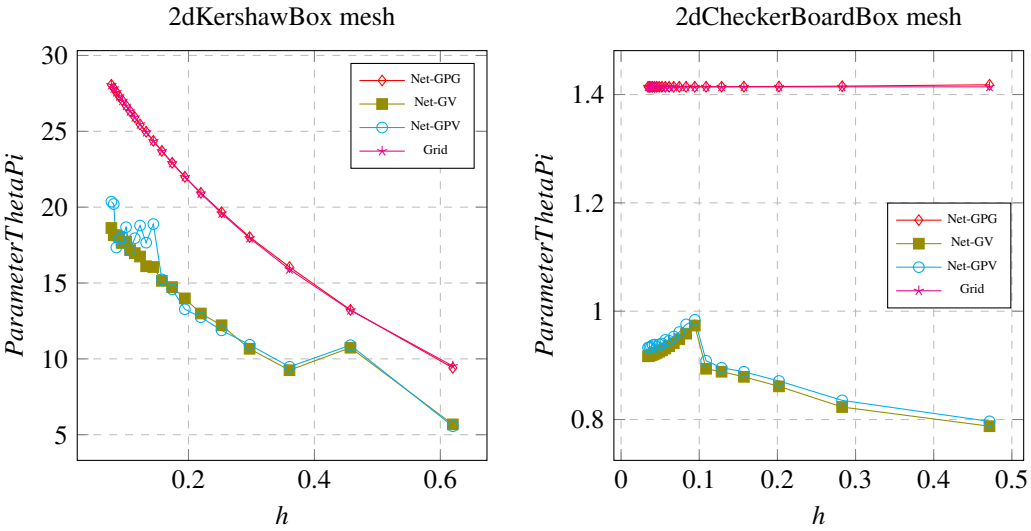

Figure 10: Quality parameter θ_Π for the *2dKershawBox* and *2dCheckerBoardBox* mesh sequences

geometry generation procedure though optimization as well as the network element method itself.

To conclude our mesh-based experiments, we consider two configurations in dimension 3. We consider the test case *Sinusoidal3D*:

$$u(x,y) = sin(\pi x)sin(\pi y)sin(\pi z) \tag{43}$$

We test this solution on a randomized box mesh sequence *3dRandomBox*, as well as on the distorted mesh sequence *3dSweep* displayed on figure 13. Meshes of this last sequence are obtained by mapping a polygonal mesh of the unit square in the xy plane to a non planar surface. In order to do so, we need to relax the constraint that u should vanish on the boundary and allow non-homogeneous Dirichlet boundary conditions, by simply taking the values of the function at interfaces belonging to $\mathcal{F}_{ext}$, i.e. setting $u_\sigma = u(\boldsymbol{x}_\sigma)\ \forall\ \sigma \in \mathcal{F}_{ext}$. Of course this is clearly out of the scope of the theory developed in the present paper, however as both u and its trace are smooth, the extension to such cases is relatively straightforward. We hope that the treatment of general boundary conditions with minimal regularity could

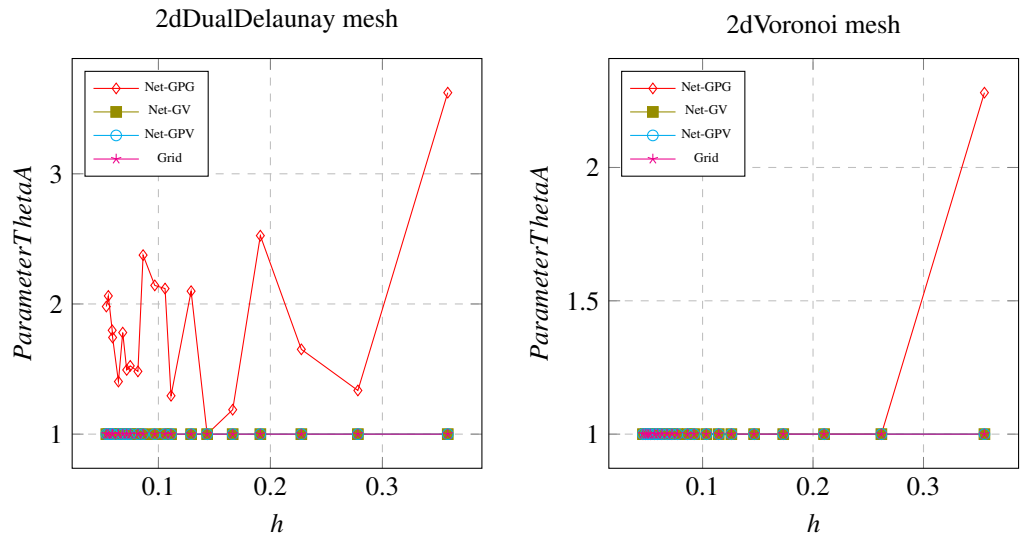

Figure 11: Quality parameter θ_A for the *2dDualDelaunay* and *2dVoronoi* mesh sequences

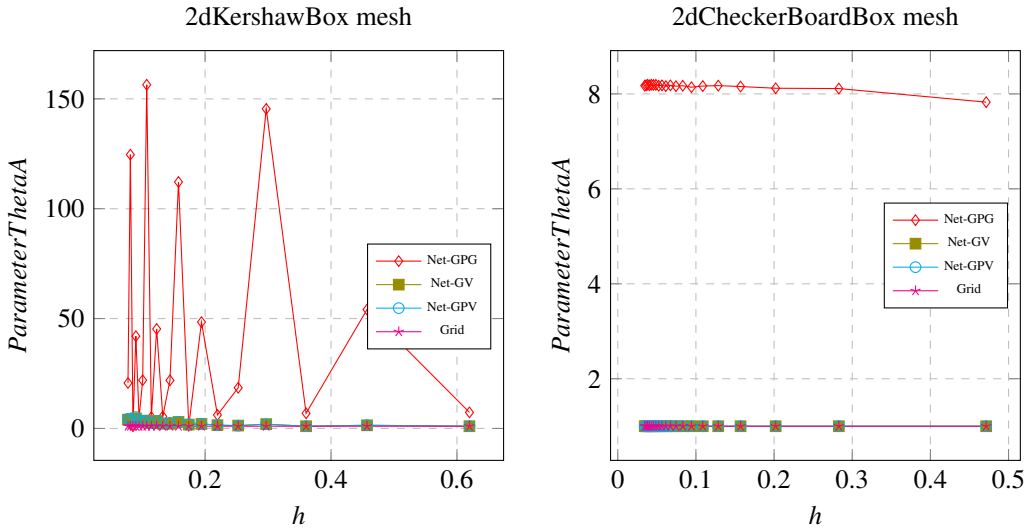

Figure 12: Quality parameter θ_A for the *2dKershawBox* and *2dCheckerBoardBox* mesh sequences

be the subject of a future paper.

Convergence curves are displayed on figure 14 and approximate convergence orders on table 2. Exactly as in dimension 2, all the mesh-based geometries lead to network element schemes with L^2 superconvergence and a behavior similar to the first order VEM. Notice that in this context, it is not surprising that the face-based *NEM-GPG* is noticeably more precise than the vertex based VEM and NEM variants: indeed, those 3d mesh sequences have much more faces than vertices, thus it is only legitimate that a considerably larger number of unknowns provide an increased precision. The quality parameters for those 3d mesh-based test cases are displayed on figures 15, 16 and 17. We see that as in dimension 2, they remain under control and in fact follow once again their equivalent on the underlying grid, further validating the network geometry validation procedure.

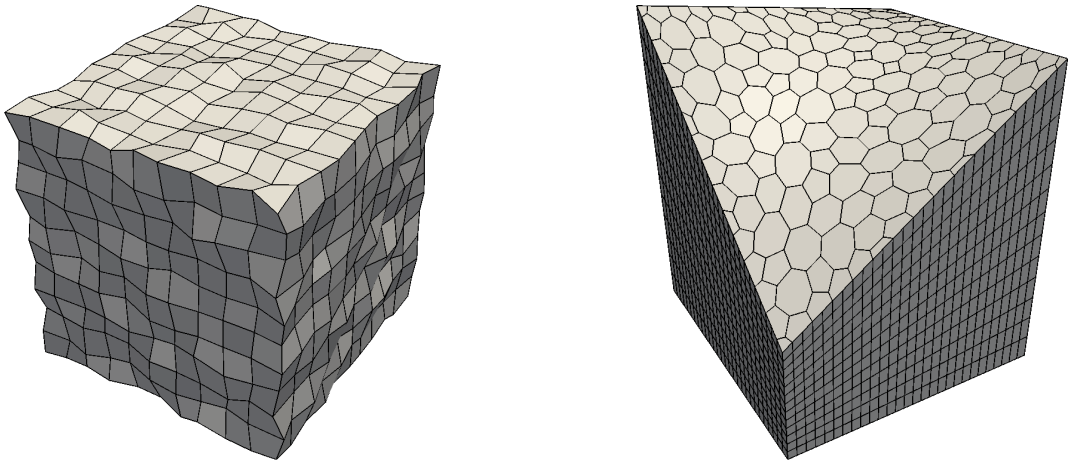

Figure 13: Example of meshes for the *3dBoxRandom* and *3dSweep* mesh sequences

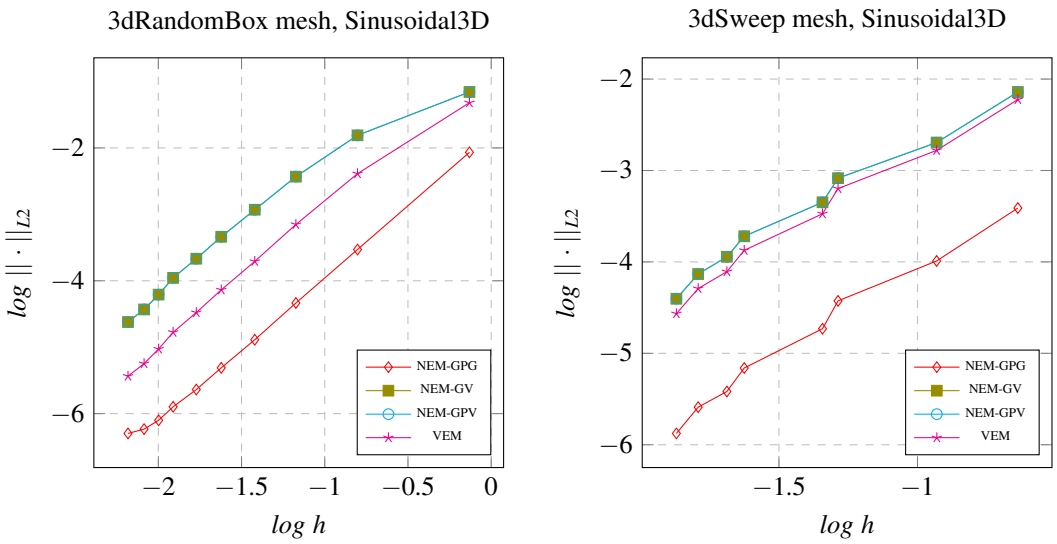

Figure 14: Convergence curves for *Sinusoidal3D* on the *3dRandomBox* and *3dSweep* mesh sequences

Table 2: Approximate orders of convergence for *Sinusoidal3D*

	3dRandomBox	*3dSweep*
VEM	2.302	2.016
NEM perturbed geometry	1.923	2.114
NEM vertex based	2.253	1.944
NEM perturbed vertex based	2.251	1.942

3.3. Generic networks

Let us consider the sectorial domain Ω of $\mathbb{R}^2$ displayed on figure 18, that is centered on the origin, of aperture angle ω and radius R, i.e. in polar coordinates

$$\Omega = \{(r,\theta) \mid 0 \leqslant r \leqslant R \text{ and } 0 \leqslant \theta \leqslant \omega\}$$

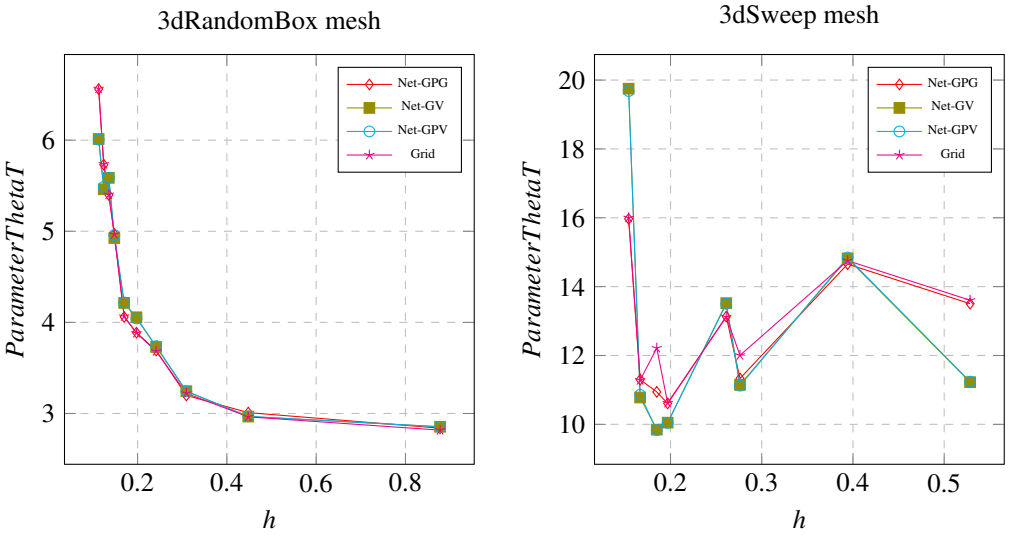

Figure 15: Quality parameter $\theta_{\mathcal{T}}^{sup}$ for the *3dRandomBox* and *3dSweep* mesh sequences

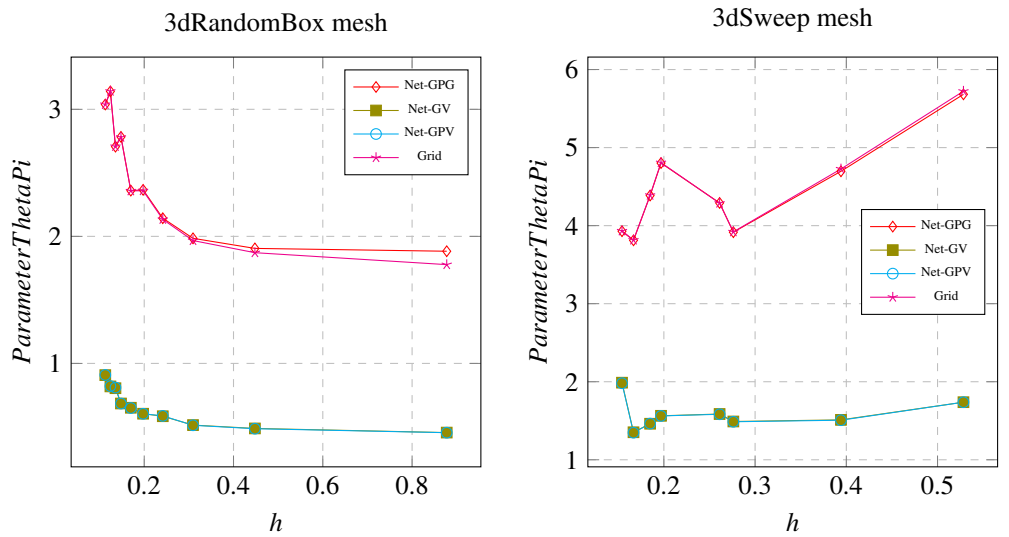

Figure 16: Quality parameter θ_{Π} for the *3dRandomBox* and *3dSweep* mesh sequences

On this domain, we define in polar coordinates the following test (named *ConvTest5*)

$$u(r,\theta) = (r-R)r^{\frac{\pi}{\omega}}sin(\frac{\pi\theta}{\omega}) \quad \text{with} \quad \Delta u(r,\theta) = \left(1+\frac{2\pi}{\omega}\right)r^{\frac{\pi}{\omega}-1}sin(\frac{\pi\theta}{\omega})$$

This test has two main interests: it will illustrate the ability of the network element method to cope with domains with curved boundaries, and allow to study its behavior when the solution regularity diminishes. Indeed, if a straightforward computation reveals that, at the origin, the first derivatives behave in $r^{\frac{\pi}{\omega}-1}$, a more tedious computation reveals that the second derivatives behave in $r^{\frac{\pi}{\omega}-2}$. Thus, while the gradient will be square integrable as soon as $\frac{\pi}{\omega} \geqslant 0$ which is always satisfied, for the second derivatives to be square integrable we need that $\frac{\pi}{\omega} \geqslant 1$. By interpolation one could then establish that for $u(r,\theta)$ to belong to $H^{1+s}(\Omega)$, we need that $s \leqslant \frac{\pi}{\omega}$, while always satisfying $u \in H_0^1(\Omega)$.

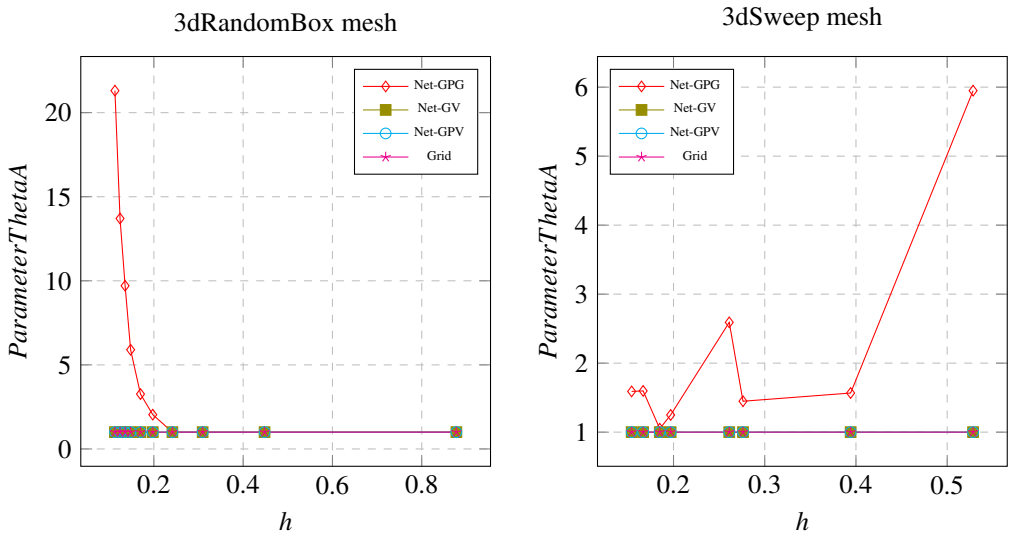

Figure 17: Quality parameter θ_A for the *3dRandomBox* and *3dSweep* mesh sequences

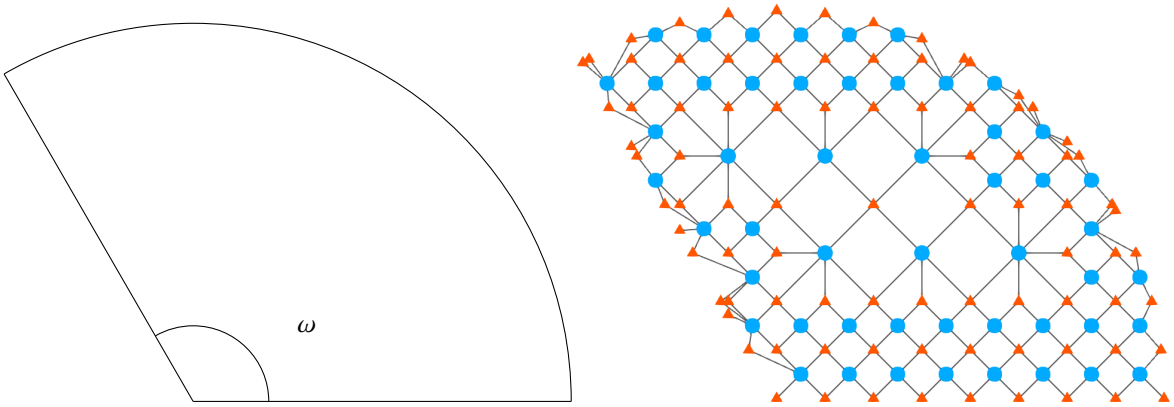

Figure 18: Sectorial domain and example of associated network (orange triangles are interfaces, blue circles are cells, lines represent the connectivity)

Table 3: Approximate orders of convergence for *AngularSector2D*

ω	120°	180°	240°
NEM	1.906	1.949	1.794

We consider three values for the aperture angle ω: 120°, 180° and 240°. The discretization networks used were generated using a cartesian matrix as basis complemented by the intersection points between the cartesian cells and the sectorial domain boundaries, as displayed on figure 18. The cartesian cells are refined in an AMR fashion near the boundaries to improve the quality parameters of the network and generated geometries. Convergence curves are displayed on figure 19 while approximate convergence rates are given on table 3. As for mesh-based networks, the measured convergence rates are coherent with L^2 superconvergence, i.e. $max(2, 2s)$ for solutions belonging to $H^{1+s}(\Omega)$. The case $\omega = 240$ is as expected the one with lowest convergence order, with an approximate rate of 1.794 coherent with the theoretical rate of 1.5 and the fact that our network generation procedure automatically refines the

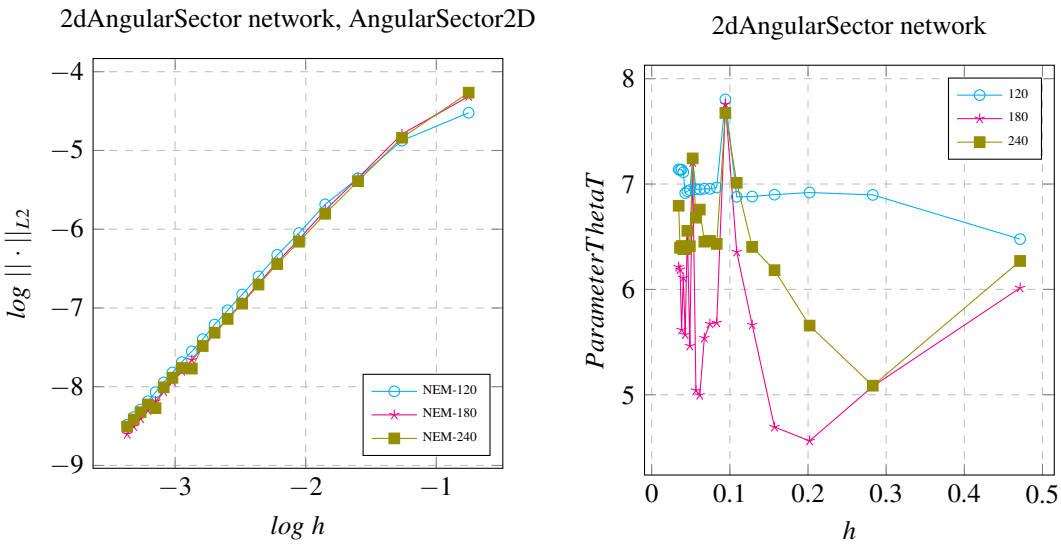

Figure 19: Convergence curves for *AngularSector2D* and parameter $\theta_{\mathcal{T}}^{sup}$ for aperture angles 120°, 180° and 240°

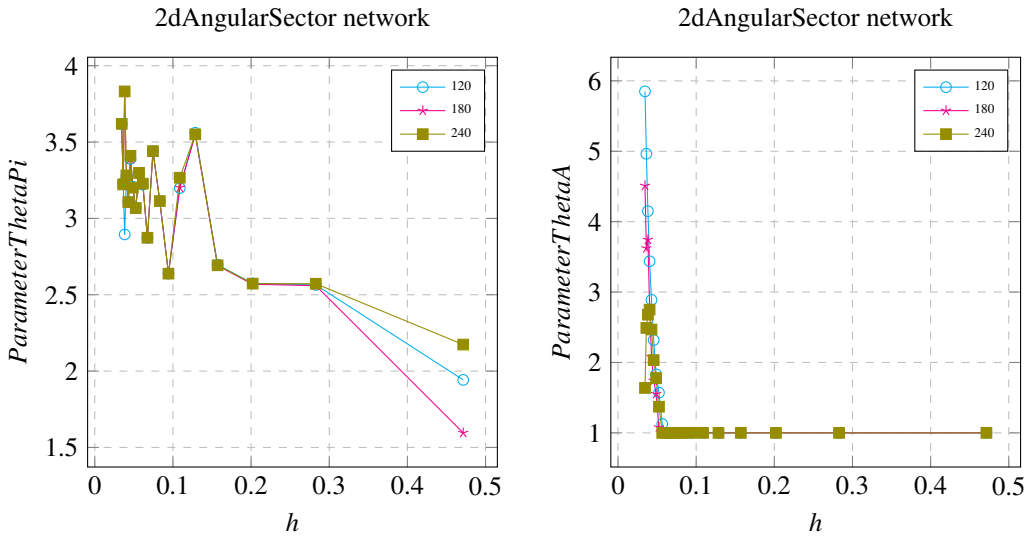

Figure 20: Quality parameters θ_{Π} and $\theta_{\mathcal{A}}$ for the *AngularSector2D* network sequences with aperture angles 120°, 180° and 240°

network near the boundary where the singularity of the solution is located. The evolution of the quality parameters with the size h is displayed on figures 19 and 20, and we see that as for mesh-based networks, no blow up is observed and they remain under control during the refinement process. In fact, most of their observed deterioration comes from the very basic handling of the boundary of the domain. Indeed, intersecting an arc with cartesian cells can lead to sets $\mathcal{F}_K$ of interfaces where all the interfaces are quite close to each other, creating a very small cell. A very basic network regularization procedure consisting in discarding the degenerate cell and agglomerating those interfaces to the closest correct cell was applied, but there most certainly remains room for improvement. In any case, we see that the results are already satisfactory for this curved domain with some non smooth analytic solutions, as we were able to generate networks and geometries with controlled quality parameters and to obtain convergent numerical solutions, resorting only to a very basic network generation procedure.

To conclude this section devoted to numerical exploration, we consider again the function

$$u(x,y) = sin(\pi x)sin(\pi y)sin(\pi z) \tag{44}$$

on a set of relatively complex 3d domains, relaxing again the constraint that u should vanish on the boundary. We consider the 3d polygonal objects displayed on figures 21 and 22, rescaled to fit inside a unitary cubic box. The corresponding .obj files were downloaded from [30] where their author has released them into the public domain. The discretization networks used were again generated using a cartesian matrix as basis complemented by the intersection points between the cartesian cells and the polygonal objects boundaries and the same regularization process was applied. The main point here is not to exhibit a generic network generation procedure, but rather to assess if even at such an early stage of the network element method and such a crude network generation procedure, we are already able to cope with domains that otherwise require quite complex meshing algorithms. Convergence curves are displayed on figure 23 while approximate convergence rates are given on table 4 and the evolution of the quality parameters with the size h is displayed on figures 23 and 24. As our network generation procedure is very simple, it generates cells for which $\hat{\rho}_K$ is quite small and the estimate $\theta_{\mathcal{T}}^{sup}$ becomes too coarse to remain relevant for those test cases. This is the reason why we replace it by

$$\widetilde{\theta_{\mathcal{T}}} = \max_{K \in \mathcal{T}} \max \left(\frac{m_K}{|B_K \cap \Omega|_{approx}}, \frac{|B_K|}{m_K} \right) < \theta_{\mathcal{T}}^{sup}$$

where $|B_K \cap \Omega|_{approx}$ is an easily computable approximation of $|B_K \cap \Omega|_{approx}$. In our experiments, we have used a basic approximation of $B_K \cap \Omega$, computed by using a cartesian discretization of the cube included in B_K and defining $|B_K \cap \Omega|_{approx}$ as the sum of the volume of the cartesian sub-boxes whose center is inside Ω.

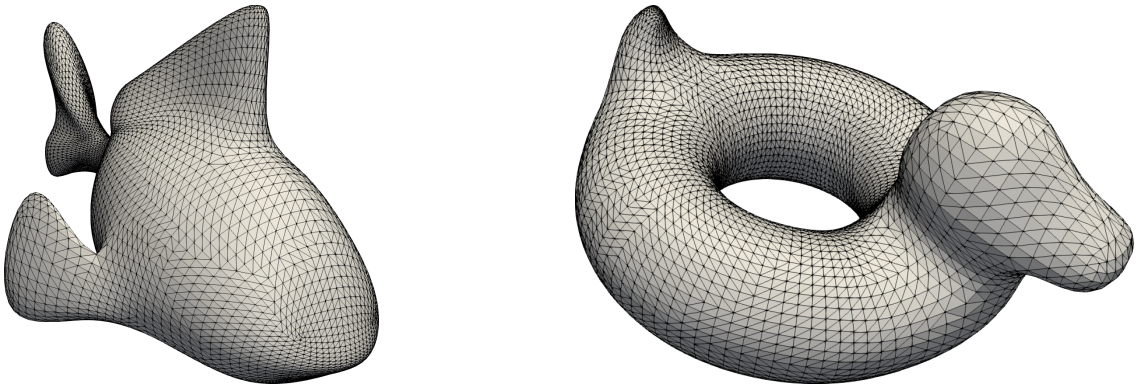

Figure 21: First and second 3d polygonal objects: fish and buoy

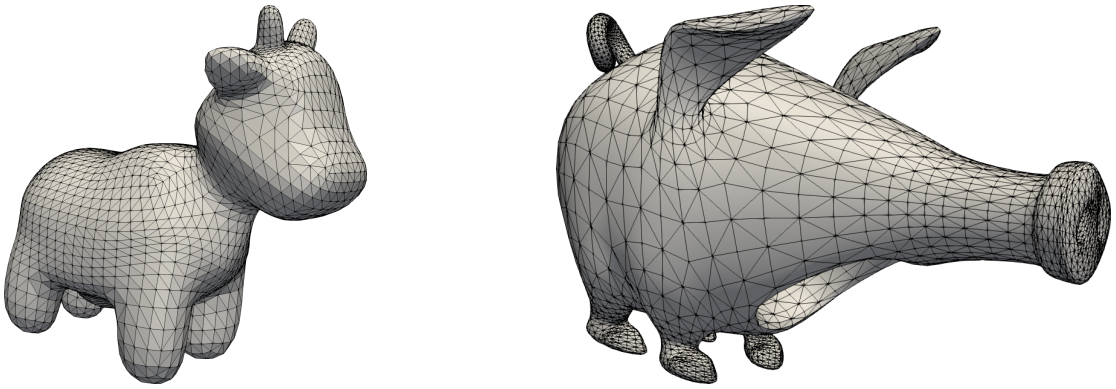

Figure 22: Third and fourth 3d polygonal objects : cow and pig

Roughly speaking, we obtain the expected L^2 superconvergence in all cases. However, the convergence curves are

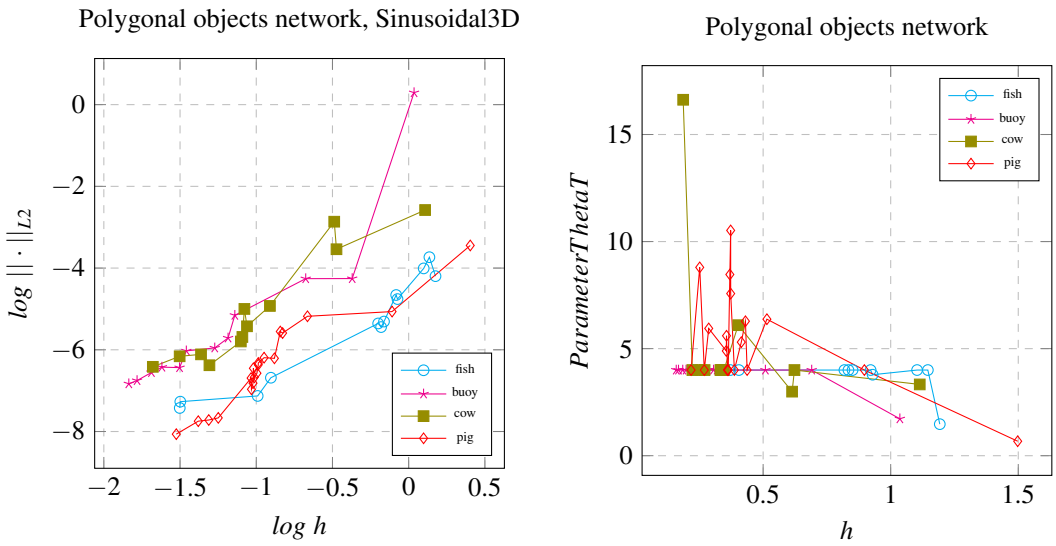

Figure 23: Convergence curves and parameter $\widetilde{\theta_{\mathcal{T}}}$ for the polygonal objects network sequences

Table 4: Approximate orders of convergence for 3d polygonal objects

	fish	*buoy*	*cow*	*pig*
NEM	2.046	1.748	2.529	2.405

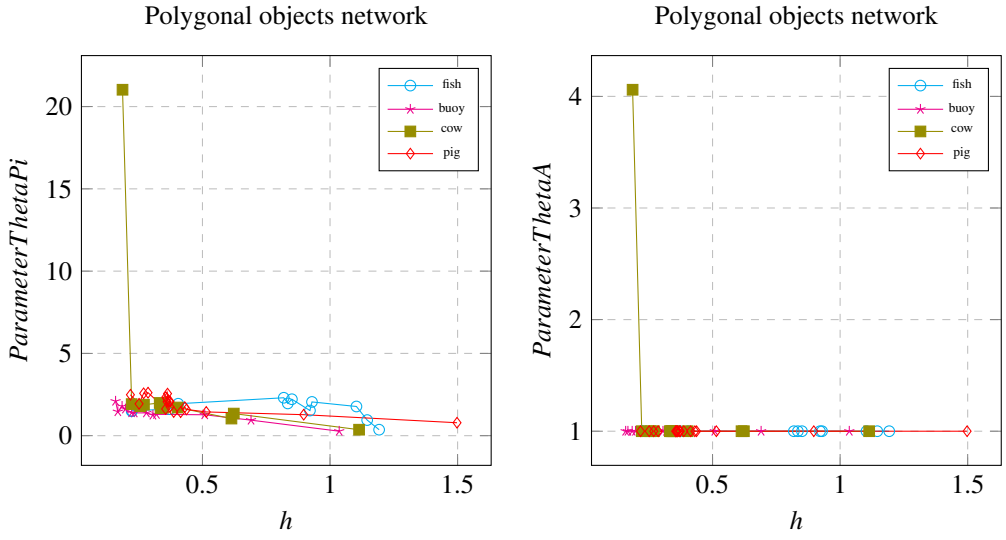

Figure 24: Quality parameters θ_{Π} and $\theta_{\mathcal{A}}$ for the polygonal objects network sequences

much less regular than in all our previous experiments, which is the clear sign that our very basic network generation procedure has generated quite irregular network sequences. Nevertheless, the network element method remains convergent, thus assessing once again its robustness to network shapes. Those experiments also emphasize the need for a more advanced network generator for complex 3d objects. However, as our very naive generator already gave acceptable results, there is hope that network generators competitive with meshing algorithms could be developed. Along with a competitive geometry computation algorithm, elaborating such a network generator is in our opinion the

key to make the network element method more than a mathematical curiosity.

3.4. Enhancing geometry generation performance

Although our main intention is to describe the network element method, let us now make some comments on the performance of geometry generation procedures in general. Solving the quadratic minimization problem (34)-(35) is a quite involved and computational time consuming process, that in general one will want to avoid. Alternative approaches exist in the literature, probably one of the most efficient in our context being the one presented in [20]. Reformulated with our notations, the idea is to prescribe the cell measures m_K through some analytic formulaes, satisfying (3) and (6) with in general no geometric error ε_Ω. Then, the cost function is reduced to

$$\mathcal{J}(\hat{\mathcal{G}}) = \sum_{i=1}^{d} \mathcal{J}_i(\hat{\mathcal{G}})$$

$$\text{with } \mathcal{J}_i(\hat{\mathcal{G}}) = \frac{\delta_{\mathcal{J}}^{\eta}}{2} \sum_{K\in\mathcal{T}} \sum_{\sigma\in\mathcal{F}_K} |\hat{\eta}_{K,\sigma}^{i}|^2 + \frac{1}{2} \sum_{K\in\mathcal{T}} \omega_{\mathcal{J},K}^{0} |\hat{\varepsilon}_{m,K}^{i,0}|^2 + \frac{1}{2} \sum_{K\in\mathcal{T}} \sum_{j=1}^{d} \omega_{\mathcal{J},K}^{1} \left|\hat{\varepsilon}_{m,K}^{1,ij}\right|^2 + \frac{1}{2} \sum_{\sigma\in\mathcal{F}_{int}} \omega_{\mathcal{J},\sigma} |\hat{\varepsilon}_{\sigma}^{i}|^2$$

Each of the $\mathcal{J}_i(\hat{\mathcal{G}})$ involving only the i-th component of each element of the network geometry, we have in fact d independent minimization problems. Moreover, as they are quadratic minimization problems with linear equality constraints, solving them finally amounts to solve d independent linear systems, corresponding to the system of Karush-Kuhn-Tucker first order optimality conditions (4)-(5)-(7) and (36)-(37)-(38)-(39) of the system corresponding to each direction i. As static condensation can once again be applied, the d resulting linear systems are of size $card(\mathcal{F}_{int})$. This does improve computational efficiency, however one has to quite arbitrarily choose the cell measures m_K. Numerical experiments reveal that the overall precision of the method is indeed impacted if one resorts to this simplified version of the geometry generation procedure, as we will briefly illustrate now. To this end, let us now consider the following choices for the cell measures

$$m_K = \frac{S_1^d}{2^d} (\hat{\rho}_K + \hat{r}_K)^d$$

with $\hat{h}_K = 2\hat{r}_K$. We normalize them afterwards such that their sum is equal to $|\Omega|$. Let us consider again the case *Sinusoidal2D* on the mesh based networks *2dDualDelaunay* and *2dKershawBox*, generated from the mesh vertices. We display the results on figures 25 and 26 as well as the approximate convergence rates in table 5. The results for the quadratic generation procedure are labeled *NEM-quad*, while the case where we prescribe the cell measures and only solve d independent linear problems is labeled *NEM-lin* (using $\delta_{\mathcal{J}}^{\eta} = 1$).

Table 5: Approximate orders of convergence for different geometry generation procedures

	2dDualDelaunay	*2dKershawBox*
NEM-quad	2.077	2.137
NEM-lin	2.333	0.630

Immediately, we see that on the *2dDualDelaunay* sequence, the results are of similar quality, and the convergence rate is roughly speaking preserved, although the convergence curve is less regular for *NEM-lin*. The situation is very different for the sequence *2dKershaw*, for which the approximation error is much worse with *NEM-lin*, and the convergence rate fails to even reach 1. This can be explained by looking the quality parameters : if both $\theta_{\mathcal{T}}$ and θ_{Π}, which we do not reproduce here, remain of similar quality for both approaches, this is not the case for $\theta_{\mathcal{A}}$. Indeed, it explodes when h goes to 0 for the *2dKershaw* sequence, and is most certainly the reason why we fail to reach convergence on this mesh-based sequence. Reproducing the same experiment on all our test cases, roughly speaking we obtain results similar to *2dDualDelaunay* for all other mesh based networks, i.e. we maintain second

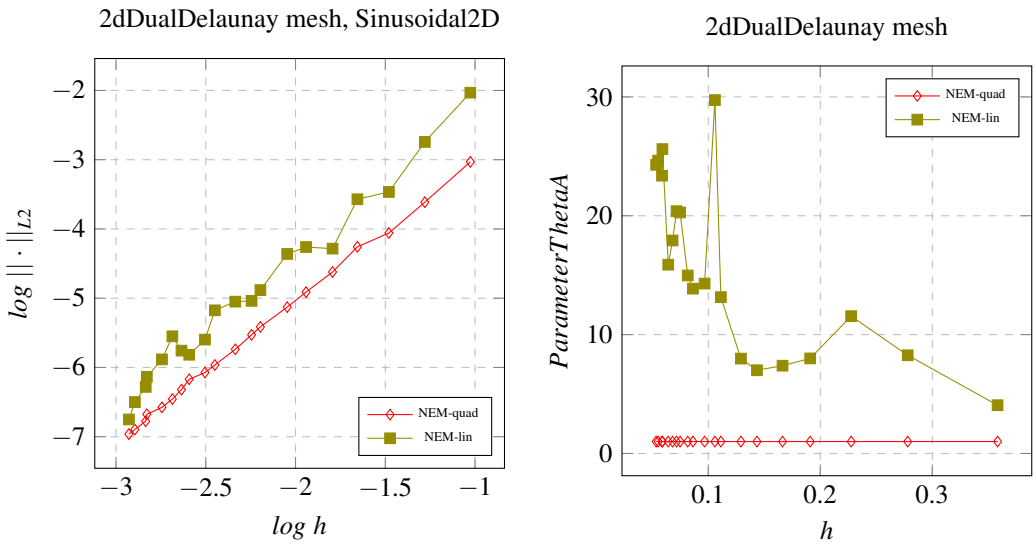

Figure 25: Convergence curves and parameter $\theta_{\mathcal{A}}$ for different geometry generation procedures

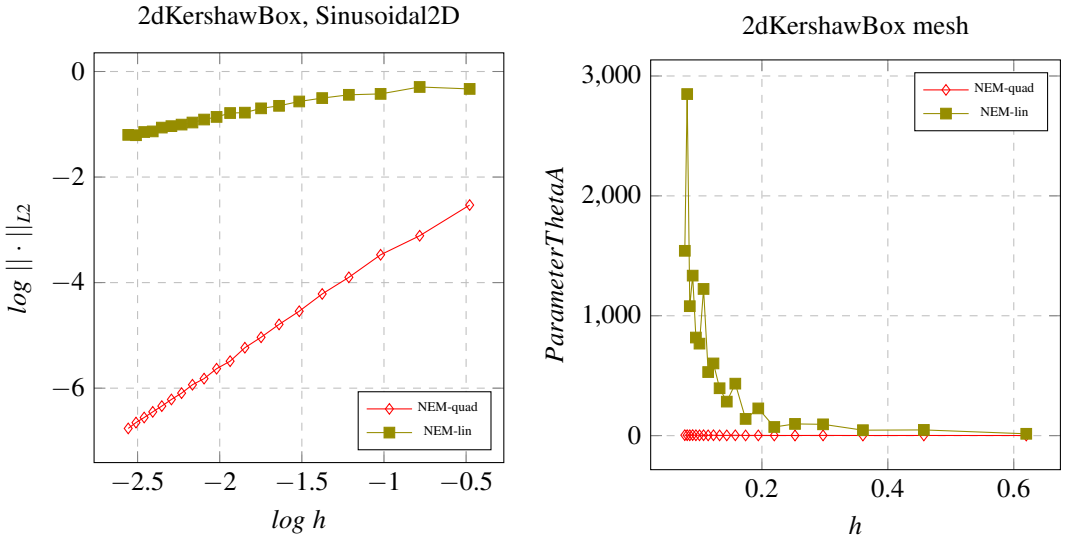

Figure 26: Convergence curves and parameter $\theta_{\mathcal{A}}$ for different geometry generation procedures

order convergence with the fast geometry generation procedure. However we have problems in maintaining second order convergence for our generic networks, with $\theta_{\mathcal{A}}$ diverging again. The impact on the approximation error is nevertheless less severe than for the case of the *2dKershaw* sequence , an order between 1 and 2 being obtained.

Some numerical exploration reveals that with the fast linear geometry generation procedure, both the choice of cost function and of the cell measures m_K have a huge impact on the quality parameter $\theta_{\mathcal{A}}$. It is thus obvious that our simple choice is far from being optimal, and that it might be feasible to always obtain second order accuracy even on the problematic *2dKershaw* sequence with the fast geometry generation procedure inspired by [20]. We have at this point no clear explanation to the fact that the quadratic optimization procedure is so efficient to control $\theta_{\mathcal{A}}$ on all our test cases. Understanding this is probably the key to derive a good cost function and a good guess for cell measures for the fast linear version. Our main objective here being to assess the potential efficiency of the network element method, we have chosen to use the quadratic minimization procedure in an attempt to obtain the best possible results for a given network configuration. Of course, as our geometry has less elements compared to the equivalent notion

of [19, 20], while having to cope with the same consistency and conservation properties, it is not surprising that its practical computation is more difficult.

4. Conclusion and perspectives

On the simplest possible model problem, we presented the principles of a network element method, a variational method on discretization networks. Based on the VEM/mimetic technologies, it reproduces their key ideas replacing the mesh by a discretization network and an associated network geometry. Numerical results in dimension 2 and 3 illustrate the good behavior and robustness of the method. The natural extension to heterogeneous and anisotropic diffusion tensors and reaction coefficients will be the subject of a future paper. The method seems fairly general and on going work concerns its extension to more complex classical problems, such as linear elasticity, Stokes flow or Maxwell's equations. An important issue that has not been considered here is also the subject of active research: the development of an automated and efficient network generation algorithm, both in dimension 2 and 3. Indeed, as the error estimates and numerical experiments reveal, the properties of the point cloud and the connectivity underlying the network strongly influences solution quality. This is not surprising as it is simply the equivalent of mesh quality in the context of discretization networks. Thus if the method is to be more than a mathematical curiosity, along with an efficient geometry computation algorithm it is essential to find a competitive network generation algorithm which could control the quality parameters of the resulting network, in particular in the long term prospect of performing adaptive refinement.

5. Acknowledgment

The author would like to thank the anonymous reviewers for their decisive comments and in particular their bibliographical remarks and advice.

References

[1] L. Beirão da Veiga, K. Lipnikov, G. Manzini, The Mimetic Finite Difference Method for Elliptic Problems, Springer, 2014.
[2] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. Marini, A. Russo, Basic principles of virtual element methods, Mathematical Models and Methods in Applied Sciences, Vol. 23(1), pp. 199-214 (2013).
[3] D. A. Di Pietro, A. Ern, S. Lemaire, An arbitrary-order and compact-stencil discretization of diffusion on general meshes based on local reconstruction operators, Computational Methods in Applied Mathematics, Vol.14(4), pp. 461-472 (2014).
[4] D. A. Di Pietro, A. Ern, Hybrid high-order methods for variable-diffusion problems on general meshes, C. R. Acad. Sci. Paris, Ser. I, Vol. 353, pp. 31-34 (2015).
[5] R. Eymard, C. Guichard, R. Herbin, Small-stencil 3d schemes for diffusive flows in porous media, ESAIM: Mathematical Modelling and Numerical Analysis, Vol. 46(2), pp. 265-290 (2011).
[6] J. Coatléven, A virtual volume method for heterogeneous and anisotropic diffusion-reaction problems on general meshes, ESAIM: Mathematical Modelling and Numerical Analysis, Vol. 51, pp. 797-824 (2017).
[7] D. A. Di Pietro, A. Ern, A hybrid high-order locking-free method for linear elasticity on general meshes, Comput. Meth. Appl. Mech. Engrg., Vol. 283, pp. 1-21 (2015).
[8] L. Beirão da Veiga, F. Brezzi, L. D. Marini, Virtual elements for linear elasticity problems, SIAM J. Numer. Anal., Vol. 51(2), pp.794-812 (2013).
[9] M. Botti, D. A. Di Pietro, P. Sochala, A hybrid high-order method for nonlinear elasticity, SIAM J. Numer. Anal., 2017, Vol. 55(6), pp. 2687-2717 (2018).
[10] L. Beirão da Veiga, C. Lovadina, D. Mora, A virtual element method for elastic and inelastic problems on polytope meshes, Computer Methods in Applied Mechanics and Engineering, Vol. 295, pp. 327-346 (2015).
[11] G. Vacca, L. Beirão da Veiga, Virtual element methods for parabolic problems on polygonal meshes, Numer Methods Partial Differential Eq, Vol.31(6), pp. 2110-2134 (2015).
[12] F. Chave, D. A. Di Pietro, L. Formaggia, A hybrid high-order method for darcy flows in fractured porous media, SIAM J. Sci. Comput., Vol. 40(2), pp. 1063-1094 (2018).
[13] L. Beirão da Veiga, C. Lovadina, G. Vacca, Divergence free virtual elements for the stokes problem on polygonal meshes, ESAIM Math. Model. Numer. Anal., Vol. 51(2), pp. 509-535 (2017).
[14] D. A. Di Pietro, A. Ern, Mathematical aspects of discontinuous Galerkin methods, Springer, 2012.
[15] A. Cangiani, G. Manzini, A. Russo, N. Sukumar, Hourglass stabilization and the virtual element method, International journal for Numerical Methods in Engineering Vol.102, pp. 404-436 (2015).
[16] J. Melenk, I. Babuška, The partition of unity finite element method: Basic theory and applications, Computer Methods in Applied Mechanics and Engineering, Vol. 139(1-4), pp. 289-314 Cambridge University Press (1996).

[17] A. Katz, A. Jameson, A meshless volume scheme, Proceedings of 19th AIAA Computational Fluid Dynamics, Fluid Dynamics and Co-located Conferences, pp. 2009-3534, American Institute of Aeronautics and Astronautics (2009).
[18] A. Katz, A. Jameson, Edge-based meshless methods for compressible viscous flow with applications to overset grids, Proceedings of the 38th Fluid Dynamics Conference and Exhibit, American Institute of Aeronautics and Astronautics (2008).
[19] N. Trask, M. Perego, P. Bochev, A high-order staggered meshless method for elliptic problems, SIAM Journal on Scientific Computing, Vol. 39(2), pp. 479-502 (2017).
[20] N. Trask, P. Bochev, M. Perego., A conservative, consistent, and scalable mesh-free mimetic method., Journal of Computational Physics, vol. 409 pp. 109-187 (2020).
[21] O. Diyankov, Uncertain grid method for numerical solution of pdes, Technical report, NeurOK Software (2008).
[22] E. K. yu Chiu, Q. Wang, R. Hu, A. Jameson, A conservative mesh-free scheme and generalized framework for conservation laws, SIAM J. Sci. Comput., Vol. 34(6), pp.2896-2916 (2012).
[23] J. M. Melenk, On Approximation in Meshless Methods, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 65–141.
[24] J.-S. Chen, M. Hillman, S.-W. Chi, Meshfree methods: Progress made after 20 years, Journal of Engineering Mechanics, Vol. 143(4) (2017).
[25] I. Babuška, U. Banerjee, J. Osborn, Survey of meshless and generalized finite element methods: A unified approach, Acta Numerica, Vol.12, pp. 1-125 (2003).
[26] J. Coatléven, Basic convergence theory for the network element method, in preparation.
[27] R. Adams, J. Fournier, Sobolev Spaces, 2nd Edition, Elsevier, 2003.
[28] P. Grisvard, Elliptic Problems in Nonsmooth Domains, Pitman Publishing Inc, Massachusetts, 1985.
[29] R. Eymard, T. Gallouët, R. Herbin, Finite volume methods, In P.G. Ciarlet and J.-L. Lions editors, Techniques of scientific computiing, Part III, Handbook of Numerical Analysis, pp. 713-1020, North-Holland, Amsterdam, 2000.
[30] K. Crane, Keenan's 3d model repository, https://www.cs.cmu.edu/~kmcrane/Projects/ModelRepository/.