



HAL
open science

On network and geometry generation for the network element method

Julien Coatléven

► **To cite this version:**

Julien Coatléven. On network and geometry generation for the network element method. 2022. hal-03870752

HAL Id: hal-03870752

<https://hal.science/hal-03870752v1>

Preprint submitted on 24 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On network and geometry generation for the network element method

Julien Coatléven^a

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

Abstract

We present efficient network and geometry generation algorithms for the network element method. After providing sufficient conditions for existence of exact geometries i.e. reproducing first order polynomials without error, we describe a basic network generation algorithm fulfilling those conditions on demand. Then we explain how one can generate network geometries by solving a linear, well-posed problem replacing the original costly non-linear approach that was previously the main computational bottleneck of the network element workflow. We finally illustrate on challenging examples the good behavior of the overall updated network element workflow, from network generation to actual computation of PDE solutions.

Keywords: Network element method, meshless methods, variational methods, virtual element method, mimetic finite differences

2010 MSC: 65N30, 65N12, 65N15

Introduction

The most classical approaches to handle discretization of partial differential equations (PDE) problems are the celebrated and mesh-based finite difference, finite element and finite volume methods, all allowing a robust discretization only on grids with simple cell geometries. Either legacy methods such as the discontinuous Galerkin method (see [1, 2] for a review) or the more modern mimetic finite differences [3, 4, 5], virtual element method ([6]) or advanced finite volume methods [7, 8, 9, 10] have been proposed as ways to handle complex cell geometries. Despite the flexibility in cell geometries brought by those new methods, the quality of the mesh remains essential to obtain good approximation of solutions. Thus, mesh generation often remains a computational bottleneck on complex geometries or when high frequency mesh adaptation is required.

This is the reason why meshless methods still represent nowadays an attractive alternative. Roughly speaking, most meshless methods fall into two main families: collocation based methods and variational methods that uses meshfree basis functions. For instance, the famous smooth particle hydrodynamics (SPH, [11, 12]), reproducing kernel particle methods (RPK, [13]), generalized finite differences (GFD, [14]), moving least-square based methods (see [15, 16]) or methods based on radial basis functions (see [17, 18, 19, 15]) are collocation methods. The main difficulty with those methods is to avoid ill-conditioning of the final system to be solved. The diffuse element method (see [20]), the element free Galerkin method (see [21, 22]), partition of unity based methods (see [23, 24]) are examples of mesh-free variational methods, for which the main difficulty is the numerical integration of those basis functions that can severely deteriorate the theoretical stability (see [15, 23]). For a review on meshless methods and of recent progress made on them, we refer the reader to [25].

A third small family of meshless methods is formed by point network based meshless methods, whose common principle is to mimic the various differential operators on the network formed by the connectivity of a point cloud, the literature generally focusing on finite volume like approaches ([26, 27, 28, 29, 30, 31]). The recently introduced network element method (NEM) is a meshless method belonging to this third family, its major originality being to provide a truly variational method with unconditional stability and a complete convergence theory (see [32, 33, 34]). Up to this point, the main drawback of the NEM is the practical computation of the so-called network geometry for which a costly constrained optimization problem was introduced in [32], nevertheless providing good quality geometries even on very distorted networks. As an attempt to alleviate the cost of this non-linear problem in [32] it was

proposed to solve instead d independent linear optimization problems, d being the dimension, with results of poor quality on too distorted networks.

The main objective of the present paper is to propose efficient solutions for both network and geometry generation for the network element method. Concerning network generation, essentially this consists in using a node placing algorithm for which a huge literature exists, among which advancing front methods seem to be some of the most promising tools. A network generator will then just need to correctly define the connectivity of the obtained point cloud. As we will mainly rely on node placing algorithms as an intermediate tool for network generation, we refer the reader to [35, 36, 37, 38, 39] and the references therein for a complete survey on node placing algorithms. Concerning geometry generation, one of the key points to control the approximation quality is to determine whether there exists “exact” geometries capable of representing exactly first order polynomials. In [32] it is established that this cannot be done for fully general networks, however here we will exhibit some sufficient conditions ensuring that such exact geometries exist. Our network generation algorithm will thus be defined in such a way that those sufficient conditions can be incorporated at the network generation stage. Then, we will explain how one can modify the non-linear optimization problem of [32] into a linear, well posed problem producing geometries of the same quality, even exact ones if the sufficient conditions are satisfied. We believe that the proposed algorithms considerably improve the overall workflow of the network element method up to the point of making it a realistic alternative to other existing meshless methods.

The paper will be organized as follows: after recalling the precise definitions of a discretization network and associated geometry, we will focus on sufficient conditions for exact geometries to exist. Then, building on those sufficient conditions and on an advancing front node generator of the literature, we will propose a basic network generation algorithm. The next section is devoted to geometry generation, where we exhibit a linear simplification of the non-linear geometry computation problem of [32], establishing its well-posedness in general. Finally, in a last section we will present some numerical results for both network and geometry generation.

1. Discretization networks and network geometry

Let $d \in \mathbb{N} \setminus \{0\}$, for any $\mathbf{x} \in \mathbb{R}^d$ and any $r > 0$, we denote $B(\mathbf{x}, r)$ the ball of radius r centered at \mathbf{x} for the euclidean norm $\|\mathbf{x}\|^2 = \sum_{i=1}^d x_i^2$. Following [32] we recall that a discretization network \mathcal{N} of the open bounded connected subset Ω of \mathbb{R}^d 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 $K = \{\mathbf{x}_K, r_K\}$, with $\mathbf{x}_K \in \mathcal{P}_{\mathcal{T}}$ 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 $\sigma = \{\mathbf{x}_\sigma, \mathcal{T}_\sigma\}$, with $\mathbf{x}_\sigma \in \mathcal{P}_{\mathcal{F}}$ and \mathcal{T}_σ 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 $\sigma \in \mathcal{F}_{ext}$, \mathbf{x}_σ is a point in $\cap_{K \in \mathcal{T}_\sigma} B(\mathbf{x}_K, r_K) \cap \partial\Omega$. The set of interior interfaces \mathcal{F}_{int} is such that for all $\sigma \in \mathcal{F}_{int}$, \mathbf{x}_σ is a point in $\cap_{K \in \mathcal{T}_\sigma} B(\mathbf{x}_K, r_K) \cap \bar{\Omega}$.
- $\bar{\Omega} \subset \bigcup_{K \in \mathcal{T}} B(\mathbf{x}_K, r_K)$. For all $(K_1, K_2) \in \mathcal{T}^2$ such that $K_1 \neq K_2$, $\mathbf{x}_{K_1} \neq \mathbf{x}_{K_2}$.
- To any $K \in \mathcal{T}$, we associate a Lipschitz open set $\bar{\Omega}_K \subset B(\mathbf{x}_K, r_K)$ such that $\bar{\Omega} \subset \bigcup_{K \in \mathcal{T}} \bar{\Omega}_K$. For any $K \in \mathcal{T}$ such that $\partial\Omega \cap \bar{\Omega}_K \neq \emptyset$, there exists $\sigma \in \mathcal{F}_{ext}$ such that $K \in \mathcal{T}_\sigma$. For any $(K, L) \in \mathcal{T}^2$ such that $\bar{\Omega}_K \cap \bar{\Omega}_L \neq \emptyset$, there exists $\sigma \in \mathcal{F}$ such that $(K, L) \subset \mathcal{T}_\sigma$.

For any $K \in \mathcal{T}$, the set $\mathcal{F}_K = \{\sigma \in \mathcal{F} \mid K \in \mathcal{T}_\sigma\}$ denotes the interfaces of K , which implies that for any $\sigma \in \mathcal{F}$, \mathcal{T}_σ denotes the cells connected to the interface σ and satisfies $\mathcal{T}_\sigma = \{K \in \mathcal{T} \mid \sigma \in \mathcal{F}_K\}$. A network is said to be admissible if for any cell $K \in \mathcal{T}$ the set $(\mathbf{x}_\sigma)_{\sigma \in \mathcal{F}_K}$ is unisolvent for first order polynomials.

Remark 1.1. The last part of the above definition differs from the original definition of [32] where it was replaced by the less general hypothesis:

- For any $K \in \mathcal{T}$ such that $\partial\Omega \cap \overline{B(\mathbf{x}_K, r_K)} \neq \emptyset$, there exists $\sigma \in \mathcal{F}_{ext}$ such that $K \in \mathcal{T}_\sigma$, and for any $(K, L) \in \mathcal{T}^2$ such that $B(\mathbf{x}_K, r_K) \cap B(\mathbf{x}_L, r_L) \neq \emptyset$, there exists $\sigma \in \mathcal{F}$ such that $(K, L) \subset \mathcal{T}_\sigma$.

The role of this hypothesis in [32] is mainly to allow to establish a discrete Poincaré's inequality. A careful look at the proof of the discrete Poincaré inequality of [32] reveals that it remains valid under the more general hypothesis we consider here. We introduce this generalization as it will exactly match the networks produced by the network generation algorithm we consider here.

1.1. Network geometry

We define a network geometry as a set of coefficients:

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

The discrete measures $(m_K)_{K \in \mathcal{T}}$ are admissible if and only if they satisfy $m_K > 0$ for all $K \in \mathcal{T}$ and

$$\sum_{K \in \mathcal{T}} m_K = |\Omega|. \quad (1)$$

The approximate consistency properties are given by

$$\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i = m_K \varepsilon_K^{0,i} \quad \text{for all } K \in \mathcal{T} \text{ and all } 1 \leq i \leq d, \quad (2)$$

and

$$\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i (x_\sigma^j - x_K^j) = m_K (\delta_{ij} + \varepsilon_K^{1,ij}) \quad \text{for all } K \in \mathcal{T} \text{ and all } 1 \leq i, j \leq d, \quad (3)$$

and the approximate compatibility (or conservation) properties by:

$$\sum_{K \in \mathcal{T}_\sigma} \eta_{K,\sigma}^i = \varepsilon_\sigma^i \quad \text{for all } \sigma \in \mathcal{F}_{int} \text{ and all } 1 \leq i \leq d. \quad (4)$$

As explained in [32], the subset

$$\mathcal{G}_r = \left((m_K)_{K \in \mathcal{T}}, (\eta_{K,\sigma}^i)_{K \in \mathcal{T}, \sigma \in \mathcal{F}_K, 1 \leq i \leq d} \right),$$

contains the coefficients needed to derive the network element operators while the subset

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

is intended to represent the approximation error we tolerate on the exact geometrical constraints. We define the constants $\theta_{\mathcal{A}} > 0$ and $p \geq 1$, both independent on h , and such that

$$|\varepsilon_K^{0,i}| \leq \theta_{\mathcal{A}} h_K^p \quad \text{and} \quad |\varepsilon_K^{1,ij}| \leq \theta_{\mathcal{A}} h_K^p \quad \text{for all } K \in \mathcal{T} \text{ and all } 1 \leq i, j \leq d, \quad (5)$$

and

$$|\varepsilon_\sigma^i| \leq \theta_{\mathcal{A}} \min_{K \in \mathcal{T}_\sigma} m_K h_K^p \quad \text{for all } \sigma \in \mathcal{F}_{int} \text{ and all } 1 \leq i \leq d. \quad (6)$$

We say that a network geometry is admissible if and only it satisfies (2)-(3)-(4) and if the family of measures is admissible. An admissible network geometry for which $\varepsilon(\mathcal{G}) = 0$ is an exact geometry, and we simply denote

$$\mathcal{G} = \mathcal{G}_r = \left((m_K)_{K \in \mathcal{T}}, (\eta_{K,\sigma}^i)_{K \in \mathcal{T}, \sigma \in \mathcal{F}_K, 1 \leq i \leq d} \right),$$

with a slight abuse of notations in this case.

2. Sufficient conditions ensuring existence of exact geometries for any set of admissible measures

From [32], we know that for an exact geometry to exist the family $(m_K)_{K \in \mathcal{T}}$ must satisfy a set of linear constraints. The aim of this section is to give sufficient conditions for this set of linear constraints to be empty, and thus for an exact geometry to exist for any family of positive $(m_K)_{K \in \mathcal{T}}$ satisfying (1). We begin by describing a quite general but impractical sufficient condition, that will be our guideline to derive more practical stronger conditions. We say that an admissible network is unisolvent for piecewise first order polynomials (H) if for any set of cell associated first order vector polynomials $\mu_K = (\mu_K^i)_{1 \leq i \leq d}$ with $\mu_K^i(\mathbf{x}) = \mu_K^{0,i} + \sum_{j=1}^d \mu_K^{1,ij}(x^j - x_K^j)$ for all $K \in \mathcal{T}$ and internal interfaces associated vector values $\mu_\sigma = (\mu_\sigma^i)_{1 \leq i \leq d}$ for all $\sigma \in \mathcal{F}_{int}$ such that:

- (H-i): $\mu_K^i(\mathbf{x}_\sigma) = \mu_K^{0,i} + \mu_K^{1,ij}(x_\sigma^j - x_K^j) = 0$ for all $\sigma \in \mathcal{F}_K \cap \mathcal{F}_{ext}$ and all $1 \leq i, j \leq d$,
- (H-ii): $\mu_K^i(\mathbf{x}_\sigma) = \mu_K^{0,i} + \mu_K^{1,ij}(x_\sigma^j - x_K^j) = -\mu_\sigma^i$ for all $\sigma \in \mathcal{F}_K \cap \mathcal{F}_{int}$ and all $1 \leq i, j \leq d$,

then $\mu_K = 0$ for all $K \in \mathcal{T}$ and $\mu_\sigma = 0$ for all $\sigma \in \mathcal{F}_{int}$. We denote $\mu = ((\mu_K)_{K \in \mathcal{T}}, (\mu_\sigma)_{\sigma \in \mathcal{F}_{int}})$ such piecewise first order polynomials in the following.

Immediately, we deduce the following existence result:

Proposition 2.1. *Let \mathcal{N} be an admissible network. Assume that \mathcal{N} is “unisolvent for piecewise first order polynomials”. Then for any admissible family $(m_K)_{K \in \mathcal{T}}$, there exists a family $(\eta_{K,\sigma})_{K \in \mathcal{T}, \sigma \in \mathcal{F}_K}$ such that \mathcal{G}_r is an exact network geometry.*

Proof. To get an exact geometry, we must find a family of $\eta_{K,\sigma}$'s such that:

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

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

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

We have:

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

As $\mathbb{A}_G^T \boldsymbol{\mu} = 0$ is equivalent to $(\mathbb{A}_G^T \boldsymbol{\mu})^T \boldsymbol{\eta} = 0$ for all $\boldsymbol{\eta}$ we see, taking one element of $\boldsymbol{\eta}$ equal to one and all the others equal to zero, that $\mathbb{A}_G^T \boldsymbol{\mu} = 0$ immediately implies that for any $K \in \mathcal{T}$ such that $\mathcal{F}_K \cap \mathcal{F}_{ext} \neq \emptyset$, we have

$$\mu_K^{0,i} + \mu_K^{1,ij} (x_\sigma^j - x_K^j) = 0 \quad \text{for all } \sigma \in \mathcal{F}_K \cap \mathcal{F}_{ext} \text{ and all } 1 \leq i, j \leq d,$$

while for any $\sigma \in \mathcal{F}_{int}$, we have

$$\mu_\sigma^i + \mu_K^{0,i} + \sum_{j=1}^d \mu_K^{1,ij} (x_\sigma^j - x_K^j) = 0 \quad \text{for any } K \in \mathcal{T}_\sigma.$$

From the hypothesis, we immediately deduce that $Ker \mathbb{A}_G^T = \{0\}$ which from Fredholm's alternative concludes the proof. \square

Now, we are going to present stronger conditions that all imply (H) and thus existence of exact geometries. We start by describing a very strong sufficient condition that will give the general idea underlying its weaker, more involved version. We say that a network \mathcal{N} has a “full connectivity” (H1) if and only if

- (H1-i): the set $(\mathbf{x}_\sigma)_{\sigma \in \mathcal{F}_{ext}}$ is unisolvent for $\mathbb{P}_1(\mathbb{R}^d)$.
- (H1-ii): for any $(K, L) \in \mathcal{T}^2$ such that $\mathcal{F}_K \cap \mathcal{F}_L \neq \emptyset$ the set $(\mathbf{x}_\sigma)_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_L}$ is unisolvent for $\mathbb{P}_1(\mathbb{R}^d)$,

Proposition 2.2. *Let \mathcal{N} be an admissible network. Assume that \mathcal{N} has a “full connectivity”. Then \mathcal{N} is unisolvent for first order polynomials and thus for any admissible family $(m_K)_{K \in \mathcal{T}}$, there exists a family $(\boldsymbol{\eta}_{K,\sigma})_{K \in \mathcal{T}, \sigma \in \mathcal{F}_K}$ such that \mathcal{G}_r is an exact network geometry.*

Proof. Assume that for any $K \in \mathcal{T}$ such that $\mathcal{F}_K \cap \mathcal{F}_{ext} \neq \emptyset$, we have

$$\mu_K^{0,i} + \mu_K^{1,ij}(x_\sigma^j - x_K^j) = 0 \quad \text{for all } \sigma \in \mathcal{F}_K \cap \mathcal{F}_{ext} \text{ and all } 1 \leq i, j \leq d.$$

This implies that the first order polynomial $\mu_K^i(\mathbf{x}) = \mu_K^{0,i} + \sum_{j=1}^d \mu_K^{1,ij}(x^j - x_K^j)$ cancels at each $(\mathbf{x}_\sigma)_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_{ext}}$. We denote $\mathcal{B} = \{K \in \mathcal{T} \mid \mathcal{F}_K \cap \mathcal{F}_{ext} \neq \emptyset\}$ the set of such boundary cells. Next, for any $\sigma \in \mathcal{F}_{int}$, we have

$$\mu_\sigma^i + \mu_K^{0,i} + \sum_{j=1}^d \mu_K^{1,ij}(x_\sigma^j - x_K^j) \quad \text{for any } K \in \mathcal{T}_\sigma,$$

thus if (K, L) are two cells such that $\mathcal{F}_K \cap \mathcal{F}_L \neq \emptyset$, using the hypothesis we deduce that for any $1 \leq i \leq d$, $\mu_K^i(\mathbf{x}) = \mu_L^i(\mathbf{x})$ as they are equal at each $(\mathbf{x}_\sigma)_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_L}$. Proceeding by induction, as Ω is assumed to be connected, using the last property in the definition of a network for each pair of cell (K, K') there exists a finite “path” of cells $(K_n)_{0 \leq n \leq M}$ such that;

$$K_0 = K, \quad K_M = K' \quad \text{and} \quad \mathcal{F}_{K_n} \cap \mathcal{F}_{K_{n+1}} \neq \emptyset \quad \text{for all } 0 \leq n \leq M-1,$$

we deduce that for any $K \in \mathcal{T}$ and any $1 \leq i \leq d$, $\mu_K^i = \mu^i \in \mathbb{P}^1(\mathbb{R}^d)$, and in particular for any $K \in \mathcal{B}$. This implies that μ^i cancels at each $(\mathbf{x}_\sigma)_{\sigma \in \mathcal{F}_{ext}}$, and is thus zero, which concludes the proof. \square

For simple geometric configurations in particular when there is no need to match the discontinuities of irregular coefficients, the above sufficient condition is easy to ensure while being relatively costly: it requires a large number of interfaces in particular when compared to mesh-based networks, which will ultimately lead to much more unknowns for the network element method. However it is useful to understand the origin of the following weaker condition, designed to handle situations where one want the network interfaces to match the discontinuity surfaces of some coefficient appearing in the problem to be solved by the network element method. We say that network has “sequentially complete connectivity” (H2) if and only if:

- (H2-i): there exists M admissible sub-networks $(\mathcal{N}^m)_{0 \leq m \leq M-1} = (\mathcal{T}^m, \mathcal{F}^m)_{0 \leq m \leq M-1}$ such that:

$$\mathcal{T} = \bigcup_{0 \leq m \leq M-1} \mathcal{T}^m \text{ with } \mathcal{T}_n \cap \mathcal{T}_m = \emptyset \text{ if } n \neq m, \quad \text{and} \quad \mathcal{F} = \bigcup_{0 \leq m \leq M-1} \mathcal{F}^m,$$

where for any $0 \leq m \leq M-1$, $\mathcal{F}^m = \mathcal{F}_{ext}^m \cup \mathcal{F}_{int}^m$ with

$$\mathcal{F}_{ext}^m = \{\sigma \in \mathcal{F}_{ext} \mid \text{card}(\mathcal{T}_\sigma \cap \mathcal{T}^m) > 0\} \cup \{\sigma \in \mathcal{F}_{int} \mid \text{card}(\mathcal{T}_\sigma) > \text{card}(\mathcal{T}_\sigma \cap \mathcal{T}^m) > 0\},$$

and

$$\mathcal{F}_{int}^m = \{\sigma \in \mathcal{F}_{int} \mid \text{card}(\mathcal{T}_\sigma) = \text{card}(\mathcal{T}_\sigma \cap \mathcal{T}^m) > 0\},$$

- (H2-ii): the set $(\mathbf{x}_\sigma)_{\sigma \in \mathcal{F}_{ext}^0 \cap \mathcal{F}_{ext}}$ is unisolvent for $\mathbb{P}_1(\mathbb{R}^d)$,

- (H2-iii): for any $K \in \mathcal{T}^0$, there exists a subset $\mathcal{B}^0(K)$ of \mathcal{T}^0 such that the set $(\mathbf{x}_\sigma)_{\sigma \in \mathcal{F}_{ext}^0(\mathcal{B}^0(K))}$ is unisolvent for $\mathbb{P}_1(\mathbb{R}^d)$, with

$$\mathcal{F}_{ext}^0(\mathcal{B}^0(K)) = \{\sigma \in \mathcal{F}_{ext}^0 \cap \mathcal{F}_{ext} \mid \text{card}(\mathcal{T}_\sigma \cap \mathcal{B}^0(K)) > 0\},$$

and for any $K' \in \mathcal{B}^0(K)$, there exists a finite path of cells $(K_n)_{0 \leq n \leq N(K, K')} \subset \mathcal{T}^0$ such that $K_0 = K$, $K_{N(K, K')} = K'$ and for all $0 \leq n < N(K, K')$, the set $(\mathbf{x}_\sigma)_{\sigma \in \mathcal{F}_{K_n} \cap \mathcal{F}_{K_{n+1}}}$ is unisolvent for $\mathbb{P}_1(\mathbb{R}^d)$.

- (H2-iv): for any $1 \leq m \leq M - 1$ and any $K \in \mathcal{T}^m$, there exists a subset $\mathcal{B}^m(K)$ of \mathcal{T}^m such that the set $(\mathbf{x}_\sigma)_{\sigma \in \mathcal{F}_{ext}^m(\mathcal{B}^m(K))}$ is unisolvent for $\mathbb{P}_1(\mathbb{R}^d)$, with

$$\mathcal{F}_{ext}^m(\mathcal{B}^m(K)) = (\mathcal{F}_{ext}^m \cap \mathcal{F}_{ext}) \cup \bigcup_{0 \leq k \leq m-1} \mathcal{F}_{ext}^{m,k}(\mathcal{B}^m(K)),$$

where

$$\mathcal{F}_{ext}^{m,k}(\mathcal{B}^m(K)) = \{\sigma \in \mathcal{F}_{ext}^m \cap \mathcal{F}_{ext}^k \mid \text{card}(\mathcal{T}_\sigma \cap \mathcal{B}^m(K)) > 0\},$$

and for any $K' \in \mathcal{B}^m(K)$, there exists a finite path of cells $(K_n)_{0 \leq n \leq N(K, K')} \subset \mathcal{T}^m$ such that $K_0 = K$, $K_{N(K, K')} = K'$ and for all $0 \leq n < N(K, K')$, the set $(\mathbf{x}_\sigma)_{\sigma \in \mathcal{F}_{K_n} \cap \mathcal{F}_{K_{n+1}}}$ is unisolvent for $\mathbb{P}_1(\mathbb{R}^d)$.

Proposition 2.3. *Let \mathcal{N} be an admissible network. Assume that \mathcal{N} has a “sequentially complete connectivity”. Then for any admissible family $(m_K)_{K \in \mathcal{T}}$, there exists $(\boldsymbol{\eta}_{K, \sigma})_{K \in \mathcal{T}, \sigma \in \mathcal{F}_K}$ such that \mathcal{G}_r is an exact network geometry.*

Proof. Using the notations and reasoning of the proof of proposition 2.2, it is clear that if $\boldsymbol{\mu}$ satisfies hypothesis (H2), for any $1 \leq i \leq d$ and any $K \in \mathcal{T}^0$ and any $L \in \mathcal{B}^0(K)$ $\mu_K^i = \mu_L^i$. As $\mathcal{F}_{ext}^0(\mathcal{B}^0(K)) \subset \mathcal{F}_{ext}$, we also get that μ_K^i cancels at \mathbf{x}_σ for all $\sigma \in \mathcal{F}_{ext}^0(\mathcal{B}^0(K))$, which using the unisolvence hypothesis imply that $\mu_K^i = 0$ for all $K \in \mathcal{T}^0$ and all $1 \leq i \leq d$. We now proceed by induction. Assume that $\mu_K^i = 0$ for all $1 \leq i \leq d$ and for all $K \in \mathcal{T}^k$ with $0 \leq k \leq m - 1$. Then, using the hypothesis we know that for any $K \in \mathcal{T}^m$ and any $L \in \mathcal{B}^m(K)$, we have using the reasoning of proposition 2.2 that $\mu_K^i = \mu_L^i$. Using the induction hypothesis, we also know that this μ_K^i cancels on $\mathcal{F}^m(\mathcal{B}^m(K))$, and is thus zero because of the unisolvence hypothesis. Thus we get $\mu_K^i = 0$ for all $1 \leq i \leq d$ and for all $K \in \mathcal{T}^m$, which concludes the proof. \square

Many other variations around the core idea of getting enough interfaces to get unisolvence between subparts of the network could probably give birth to other interesting sets of sufficient conditions. In fact, among the remaining geometrical configurations that cannot be handled easily by the kind of sufficient conditions we have described, we believe that the most important ones are handled by the following weaker existence result, that handles the situation where Ω is roughly speaking subdivided into polygonal subsets with only subsets of hyperplanes separating each subpart:

Proposition 2.4. *Assume that for some $N_S > 0$,*

$$\bar{\Omega} = \bigcup_{0 \leq i \leq N_S} \bar{\Omega}_i,$$

where for any $1 \leq i, j \leq N_S$, $i \neq j$, $\partial\Omega_i \cap \partial\Omega_j$ is included in an hyperplane of \mathbb{R}^d . For any $1 \leq i \leq N_S$, we assume that we are given a network \mathcal{N}_i satisfying (H). Moreover, we assume that for any $1 \leq i \leq N_S$ and any $\sigma \in \mathcal{F}_{ext, i}$, either $\mathbf{x}_\sigma \in \partial\Omega \cap \partial\Omega_i$, or $\mathbf{x}_\sigma \in \partial\Omega_i \cap \partial\Omega_j$ for some $1 \leq j \leq N_S$ with $j \neq i$, and in this case there exists $\sigma' \in \mathcal{F}_{ext, j}$ such that $\mathbf{x}_\sigma = \mathbf{x}_{\sigma'}$. We also assume that for any $1 \leq i, j \leq N_S$, $\mathcal{F}_{ext, i} \cap \mathcal{F}_{ext, j}$ is non empty if and only if $\partial\Omega_i \cap \partial\Omega_j$ is non empty and in this case it is unisolvent for first order polynomials of the $(d - 1)$ dimensional hyperplane containing $\partial\Omega_i \cap \partial\Omega_j$. Finally, we assume that for any $1 \leq i \leq N_S$ and any $K \in \mathcal{T}_i$, $\mathbf{x}_K \in \dot{\Omega}_i$. A network $\mathcal{N} = (\mathcal{T}, \mathcal{F})$ for Ω is then defined as the union of the networks associated to each Ω_i :

$$\mathcal{T} = \bigcup_{1 \leq i \leq N_S} \mathcal{T}_i \quad \mathcal{F} = \bigcup_{1 \leq i \leq N_S} \mathcal{F}_i,$$

where interfaces are considered identical and thus merged if their geometrical locations are identical. Moreover for any $1 \leq i \leq N_S$ and any $\sigma \in \mathcal{F}_{ext, i} \cap \mathcal{F}_{ext}$, \mathbf{x}_σ is the barycenter of a planar subface of a decomposition (a surface

mesh) of one of the planar faces of $\partial\Omega_i$. Such a network \mathcal{N} is said to be “piecewise unisolvent”. If the family of measures $(m_K)_{K \in \mathcal{T}}$ is admissible and satisfies:

$$\sum_{K \in \mathcal{T}_i} m_K = |\Omega_i|,$$

then there exists $(\eta_{K,\sigma})_{K \in \mathcal{T}, \sigma \in \mathcal{F}_K}$ such that \mathcal{G}_r is an exact network geometry.

Proof. Using once again the notations and reasoning of the proof of proposition 2.2, it is clear that if $\boldsymbol{\mu} \in \text{Ker} \mathbb{A}_G^T$ (or equivalently satisfies the hypothesis of (H)), for any $1 \leq i \leq d$ and any $(K, L) \in \mathcal{T}^m$ then $\mu_K^i = \mu_L^i = \mu_m^i$. Moreover, we have $\mu_m^i = \mu_n^i$ on $\partial\Omega_m \cap \partial\Omega_n$ for any $1 \leq m, n \leq N_s$ such that $\partial\Omega_m \cap \partial\Omega_n \neq \emptyset$. Then, the compatibility condition for the family m_K is given by:

$$\sum_{m=1}^{N_s} \sum_{K \in \mathcal{T}_i} m_K \text{div}(\boldsymbol{\mu}_m) = \sum_{m=1}^{N_s} \left(\sum_{K \in \mathcal{T}_i} m_K \right) \text{div}(\boldsymbol{\mu}_m) = \sum_{m=1}^{N_s} |\Omega_m| \text{div}(\boldsymbol{\mu}_m) = \sum_{m=1}^{N_s} \int_{\Omega_m} \text{div}(\boldsymbol{\mu}_m) = \sum_{m=1}^{N_s} \int_{\partial\Omega_m} \boldsymbol{\mu}_m \cdot \mathbf{n}_{\partial\Omega_m}.$$

Thanks to the continuity condition on the intersections $\partial\Omega_m \cap \partial\Omega_n$, this leads to

$$\sum_{m=1}^{N_s} \sum_{K \in \mathcal{T}_i} m_K \left(\sum_{i=1}^d \mu_K^{1,ii} \right) = \sum_{m=1}^{N_s} \sum_{K \in \mathcal{T}_i} m_K \text{div}(\boldsymbol{\mu}_m) = \sum_{m=1}^{N_s} \int_{\partial\Omega_m \cap \partial\Omega} \boldsymbol{\mu}_m \cdot \mathbf{n}_{\partial\Omega_m} = 0.$$

Thus, the family of measures $(m_K)_{K \in \mathcal{T}}$ belong to $\text{Im} \mathbb{A}_G = \text{Ker} \mathbb{A}_G^\perp$, and there exists an associated exact network geometry. \square

This last result defines the most general sufficient condition we will describe here. We believe that this covers most practical situations, curved boundaries being paradoxically simpler to handle through the “sequentially complete connectivity” requirement.

The above results are stronger than an existence result, in the sense that we can choose the family of measures $(m_K)_{K \in \mathcal{T}}$ (or with very little constraints in the “piecewise unisolvent” case). Contrary to the geometries of [32], there is no need to check that those measures satisfy a set of quite abstract linear constraints to ensure existence of an exact geometry.

We conclude this section by considering a special class of admissible geometries, namely admissible geometries that are exactly compatible for which $\boldsymbol{\varepsilon}_\sigma = 0$ for all $\sigma \in \mathcal{F}_{int}$.

Proposition 2.5. *Let \mathcal{N} be an admissible network. Then there exists an admissible network geometry that is exactly compatible.*

Proof. This is a direct adaptation of the proof of existence of admissible geometries of [32]. We have for the $\eta_{K,\sigma}$ ’s, the $\varepsilon_K^{1,ij}$ ’s, the ε_K^0 and the ε_σ

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

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

$$\mathbf{g} = \left((\eta_{K,\sigma})_{K \in \mathcal{T}, \sigma \in \mathcal{F}_K}, (\varepsilon_{m,K}^{0,i})_{K \in \mathcal{T}, 1 \leq i \leq d}, (\varepsilon_{m,K}^{1,ij})_{K \in \mathcal{T}, 1 \leq i, j \leq d} \right)$$

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

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

We have:

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

As $\mathbb{A}_G^T \boldsymbol{\mu} = 0$ is equivalent to $(\mathbb{A}_G^T \boldsymbol{\mu})^T \mathbf{g} = 0$ for all \mathbf{g} we see, taking one element of \mathbf{g} equal to one and all the others equal to zero, that $\mathbb{A}_G^T \boldsymbol{\mu} = 0$ immediately implies for all $1 \leq i, j \leq d$ that $\mu_K^{1,ij} = 0$ and $\mu_K^{0,i} = 0$ for all $K \in \mathcal{T}$, as well as $\mu_\sigma^i + \mu_K^{0,i} + \sum_{j=1}^d \mu_K^{1,ij} (x_\sigma^j - x_K^j) = 0$ for all $K \in \mathcal{T}$ and all $\sigma \in \mathcal{F}_K \cap \mathcal{F}_{int}$, and thus also $\mu_\sigma^i = 0$ for all $\sigma \in \mathcal{F}_{int}$. Thus $\mathbb{A}_G^T \boldsymbol{\mu} = 0$ implies $\boldsymbol{\mu} = 0$, which concludes the proof. \square

3. A network generation algorithm

In this section we propose a simple yet computationally efficient network generation algorithm. It heavily relies on existing state of the art (or close to) advancing front node generation algorithms of the literature. As we will mainly use these node generation algorithms as an essential but intermediate tool, we have chosen to focus on the easy to implement algorithm of Slak and Kosec [35, 36], although other algorithms could certainly be used (in particular other advancing front node generators, see [37, 38, 39]). The key idea is to start from a point sampling of the geometry of the domain including the interior discontinuity surfaces, defining a first set of interfaces. Then, using those geometry linked interfaces, we use the node generation algorithm of Slak and Kosec to sample the domain interior with cell nodes. Finally, we propose a basic interior interface generation algorithm to complete the network generation, with eventually an interface enrichment step to fulfill hypothesis (H).

In all this section we will consider that:

$$\bar{\Omega} = \bigcup_{0 \leq i \leq N_S} \bar{\Omega}_i,$$

where $\Omega_i \cap \Omega_j = \emptyset$ if $i \neq j$, and for any $0 \leq i \leq N_S$. The sets $(\partial\Omega_i \cap \partial\Omega_j) \setminus \partial\Omega$ represents the discontinuity surfaces to be captured by the discretization. In other words, the initial point sampling X_0 is assumed to sample the boundary of each $\partial\Omega_i$, and not only $\partial\Omega$.

3.1. The advancing front node generation algorithm of Slak and Kosec

We recall briefly the principles of the node generation algorithm of Slak and Kosec ([35]). The algorithm requires a spacing function h , a list of starting nodes X_0 and a spatial search structure that allows fast identification of the nearest node to the considered node. A node list X is initialized with the starting nodes X_0 , waiting to be processed. At each iteration i , node p_i is processed: candidates nodes $c_{i,j}$ are generated on a sphere with center p_i and radius $r_i = h(p_i)$. Candidates nodes $c_{i,j}$ outside of the domain or too close to already existing nodes are rejected and remaining candidates are added to X . The iteration continues until all nodes of the growing list X have been processed and X does not grows anymore. The candidate nodes set C_i is obtained from a fixed discretization of the unit ball, translated to p_i , scaled by r_i , and to which a random rotation is applied globally. For $d = 2$ we follow [35] and the discretization of the unit ball is $C_{unit}(k) = \{(\cos \theta, \sin \theta) \mid \theta \in \{0, \theta_0, 2\theta_0, \dots, (k-1)\theta_0\}, \theta_0 = 2\pi/k\}$. In dimension 3, we set $C_{unit}(k, p) = \{(\cos \theta \sin \varphi, \sin \theta \sin \varphi, \cos \varphi) \mid \theta \in \{0, \theta_0, 2\theta_0, \dots, (k-1)\theta_0\}, \theta_0 = 2\pi/k \text{ and } \varphi \in \{0, \varphi_0, 2\varphi_0, \dots, (p-1)\varphi_0\}, \varphi_0 = 2\pi/p\}$. The original implementation of [35] with a generic spatial search structure is

Algorithm 1 Node generation algorithm of Slak and Kosec (reproduced from [35])

Input: Domain Ω and its dimension d .
Input: A nodal spacing function $h : \Omega \subset \mathbb{R}^d \mapsto (0, +\infty)$.
Input: A list of starting points X_0 .
Output: A list of points in Ω distributed according to spacing function h .

```

1: function PNP( $\Omega, h, X$ ):
2: SEARCHSTRUCTINIT( $X$ ).                                ▷ Initialize spatial search structure on points  $X$ .
3:  $i \leftarrow 0$ .                                         ▷ Current node index.
4: while  $i < |X|$  do                                     ▷ Until the queue is not empty.
5:    $p_i \leftarrow X[i]$                                    ▷ Dequeue current point.
6:    $r_i \leftarrow h(p_i)$                                 ▷ Compute its nodal spacing.
7:   for each  $c_{i,j}$  in CANDIDATES( $p_i, r_i$ ) do         ▷ Loop through candidates.
8:     if  $c_{i,j} \in \Omega$  then                             ▷ Discard candidates outside the domain.
9:        $n_{i,j} \leftarrow$  SEARCHSTRUCTCLOSEST( $c_{i,j}$ )      ▷ Find nearest node for proximity test.
10:      if  $\|c_{i,j} - n_{i,j}\| \geq r_i$  then                 ▷ Test that  $c_{i,j}$  is not too close to other nodes.
11:        APPEND( $X, c_{i,j}$ )                                ▷ Enqueue  $c_{i,j}$  as the last element of  $X$ .
12:        SEARCHSTRUCTINSERT( $c_{i,j}$ )                       ▷ Insert  $c_{i,j}$  into the spatial search structure.
13:      end if
14:    end if
15:  end do
16:   $i \leftarrow i + 1$                                     ▷ Move to the next non-expanded node.
17: end while
18: return  $X$ 
19: end function

```

presented on algorithm 1, reproduced from [35].

In [35] a kd-tree is used for the spatial search structure. Here we rely on a virtual cartesian mesh of a box containing Ω : using $h_{\max} = 2 \sup_{x \in \Omega} h(x)$, we virtually decompose a square box containing Ω into square cells with side length h_{\max} . In practice this is done by generating a hash-map, the hash-code of a node being the d integer indexes of the virtual cartesian cell containing it. Using those cartesian integer coordinates, the search for the nearest node simply requires to loop over the nodes inside the cartesian cell containing the current node and the nodes contained in its immediate cartesian neighbors, which are easily identified through the hash-code of the current node.

3.2. An Interface generation algorithm

In this subsection we assume that we have initialized our interface point list $\mathcal{P}_{\mathcal{T}_0}$ with the elements of X_0 and that our cell list is given by $\mathcal{P}_{\mathcal{T}} = X \setminus X_0$, after applying the node generation algorithm of Slak and Kosec. We will now explain how to generate the missing “interior” interfaces in a post-processing step.

We start by associating a radius to each cell node. For any $\mathbf{x}_K \in \mathcal{T}$, we find r_K such that:

$$\text{card}(N_K(r_K)) \geq d + 1 \quad \text{with} \quad N_K(r) = \{L \in \mathcal{T} \mid B(\mathbf{x}_L, r_L) \cap B(\mathbf{x}_K, r) \neq \emptyset \text{ and } (\mathbf{x}_K, \mathbf{x}_L) \cap \partial\Omega = \emptyset\}$$

Using the cartesian based spatial search structure that we have used for implementing the algorithm of Slak and Kosec, it is easy to construct $N_K(r)$: indeed, for any element of $\mathcal{P}_{\mathcal{T}}$, given a radius r we can easily generate $N_K(r)$. Indeed, assume that at some step a candidate radius is already associated to each cell node. The cartesian cells that might intersect the ball $B(\mathbf{x}_K, r)$ can be quickly identified as having cartesian index distances from the virtual cartesian cell containing \mathbf{x}_K below $(h_{\max}/r) + 1$ in all directions, and it just remains to loop over the cell nodes contained in those cartesian cells to generate $N_K(r)$, using the radius currently associated to each cell node. This defines a function $N_K(r) = \text{SEARCHSTRUCTNEIGHBOURS}(\mathbf{x}_K, r)$. The principle of our cell radius generation algorithm is very simple: given $\alpha > 0$, we initialize the radii by setting $r_K = \alpha h(\mathbf{x}_K)$ for all cell nodes of $\mathcal{P}_{\mathcal{T}}$. Then looping over cell nodes, for each $K \in \mathcal{T}$ initialize $r = r_K$ and increase r by a factor $\tau > 1$ until $\text{card}(N_K(r)) \geq P$ for some prescribed integer value $P \geq d + 1$ (in practice we use $\tau = 1.1$). This implementation is presented in algorithm 2. Roughly

speaking, our interface generation step tries to identify P neighbors for each cell $K \in \mathcal{T}$, with $P \geq d + 1$ to obtain an admissible network. This is nevertheless slightly different from a P nearest neighbor algorithm: indeed, the fact that we consider ball intersections is much more likely to produce a good spatial distribution of neighbors, with at least one neighbor in each spatial direction which is the key point for network admissibility. The role of the ratio α is to compensate for the fact the algorithm of Slak and Kosec ensures a minimal spacing between nodes and not a maximal spacing. In principle, one could start with $r_K = \frac{1}{2}h(\mathbf{x}_K)$ however we found that choosing a larger value much closer to one (in practice $\alpha = 0.95$) can greatly accelerate the interface generation step while maintaining a reasonable number of interfaces.

Algorithm 2 Cell radius generation algorithm

Input: The set $\mathcal{P}_{\mathcal{T}}$, the dimension d , the ratio α and the integer threshold $P \geq d + 1$
Input: The nodal spacing function $h : \Omega \subset \mathbb{R}^d \mapsto (0, +\infty)$.
Output: The radii $(r_K)_{K \in \mathcal{T}}$ associated with the element of $\mathcal{P}_{\mathcal{T}}$.

```

1: function RAD( $\mathcal{P}_{\mathcal{T}}, h, \alpha, P$ ):
2:   for each  $K \in \mathcal{T}$  do ▷ Loop through cell points.
3:      $r_K \leftarrow \alpha h(\mathbf{x}_K)$  ▷ Initialize cell radius with nodal spacing function.
4:   end do
5:   for each  $K \in \mathcal{T}$  do ▷ Loop through cell points.
6:      $r \leftarrow r_K$  ▷ Initialize candidate cell radius.
7:      $N_K(r) = \text{SEARCHSTRUCTNEIGHBOURS}(\mathbf{x}_K, r)$  ▷ Find neighbors using their current size  $r_L$ .
8:     while  $\text{card}(N_K(r)) < P$  do ▷ Until the radius is large enough.
9:        $r \leftarrow r * \tau$  ▷ Increase candidate radius.
10:       $N_K(r) = \text{SEARCHSTRUCTNEIGHBOURS}(\mathbf{x}_K, r)$  ▷ Update neighbors using their current size  $r_L$ .
11:    end while
12:     $r_K \leftarrow r$  ▷ Update current cell radius.
13:  end do
14:  return  $(r_K)_{K \in \mathcal{T}}$ 
15: end function

```

Now, the sets \mathcal{T}_{σ} for the interfaces σ of \mathcal{F}_0 associated to X_0 are defined as follows: if $\mathbf{x}_{\sigma} \in \partial\Omega$, i.e. for a boundary interface, \mathcal{T}_{σ} contains only the nearest cell node, otherwise it contains the two nearest cell nodes lying on different sides of the discontinuity surface the interface is sampling. We then set $\mathcal{P}_{\mathcal{F}} = \mathcal{P}_{\mathcal{F}_0} \cup \mathcal{P}_{\mathcal{F}_1}$, where $\mathcal{P}_{\mathcal{F}_1}$ represents the sets of “interior” interfaces. Interfaces of $\mathcal{P}_{\mathcal{F}_1}$ are generated in the most natural manner: for any pair $(K, L) \in \mathcal{T} \times \mathcal{T}$, $K \neq L$, if $B(\mathbf{x}_K, r_K) \cap B(\mathbf{x}_L, r_L) \neq \emptyset$ and $(\mathbf{x}_K, \mathbf{x}_L) \cap \partial\Omega = \emptyset$, we generate an interface σ with $\mathcal{T}_{\sigma} = \{K, L\}$ and $\mathbf{x}_{\sigma} = \frac{1}{2}(\mathbf{x}_K + \mathbf{x}_L)$. Using the fast identification of cell neighbors $N_K(r_K) = \text{SEARCHSTRUCTNEIGHBOURS}(\mathbf{x}_K, r_K)$, this can be done very efficiently. A possible implementation of this interface generation procedure is presented in algorithm 3.

3.3. Interface enrichment procedures

The above interface generation procedure is designed to ensure that the obtained network is admissible. In this subsection, we briefly explain a basic procedure to enrich the set of interfaces \mathcal{F} to match the requirements of the sufficient condition (H). The starting point is to fix a prescribed number of common interfaces between two connected cells at a value $n_E \leq d + 1$. Then, the most simple and natural enrichment procedure is a brute force enrichment: for any interface $\sigma \in \mathcal{F}_1$, if $n_E = d + 1$ we add d interfaces to $\mathcal{P}_{\mathcal{F}_1}$ as perturbations of \mathbf{x}_{σ} defined by:

$$\mathbf{x}_{\sigma,i} = \mathbf{x}_{\sigma} + \delta \rho h(\mathbf{x}_{\sigma}) \mathbf{e}_i \quad (7)$$

where $\rho = 0.1$ and δ is a random number between 0.5 and 1.0. When $n_E < d + 1$, we randomly choose $n_E - 1$ directions and apply (7) in each of the selected directions. The main advantage of this procedure is that it is extremely fast. Its obvious drawback is that it noticeably increases the total interface number. Notice that if choosing $n_E = d + 1$ ensures that (H1) is satisfied and thus also (H), for many practical configurations with no enrichment or with a lower value for n_E (typically $n_E = d$), hypothesis (H) will be satisfied. Indeed, imagine for instance a network defined from

Algorithm 3 Interface generation algorithm

Input: The set $\mathcal{P}_{\mathcal{T}}$ and the cell radii $(r_K)_{K \in \mathcal{T}}$.

Output: The set \mathcal{F}_1 of interior interfaces.

```
1: function INTERF( $\mathcal{P}_{\mathcal{T}}, (r_K)_{K \in \mathcal{T}}$ ):
2:  $\mathcal{F}_1 = \emptyset$ . ▷ Initialize interior interfaces.
3: for each  $K \in \mathcal{T}$  do ▷ Loop through cells.
4:    $N_K(r_K) = \text{SEARCHSTRUCTNEIGHBOURS}(\mathbf{x}_K, r_K)$  ▷ Find neighbors.
5:   for each  $L \in N_K(r_K)$  do ▷ Loop through neighboring cells.
6:      $\sigma = \{\mathbf{x}_\sigma, \mathcal{T}_\sigma\}$  with  $\mathbf{x}_\sigma = \frac{1}{2}(\mathbf{x}_K + \mathbf{x}_L)$  and  $\mathcal{T}_\sigma = \{K, L\}$  ▷ Generate interface.
7:      $\mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup \sigma$  ▷ Add interface to  $\mathcal{F}_1$ .
8:   end do
9: end do
10:  $\mathcal{F}_1 = \text{UNIQUE}(\mathcal{F}_1)$  ▷ Suppress repetitions.
11: return  $\mathcal{F}_1$ 
12: end function
```

a cartesian mesh, with interfaces defined from the mesh vertices. Then, we have exactly d common interfaces between two cells, and not $d + 1$. It is however easy to show that (H) holds nevertheless for such cartesian networks.

3.4. Summary of the network generation procedure

We summarize in algorithm 4 our network generation procedure: starting from a geometry point sampling X_0 and a size function h , we first generate cell nodes using the node generation algorithm of Slak and Kosec. Next, we generate cell radii using algorithm 2. Then we generate interior interfaces from direct ball intersections (algorithm 3) and finally we enrich the set of interior interfaces if needed.

Algorithm 4 Network generation algorithm

Input: Domain Ω and its dimension d , the ratio α and the integer threshold $P \geq d + 1$

Input: A nodal spacing function $h : \Omega \subset \mathbb{R}^d \mapsto (0, +\infty)$.

Input: A list of starting points X_0 corresponding to the geometry discretization.

Input: A flag e indicating if we should enrich and an enrichment parameter n_E

Output: The network \mathcal{N} .

```
1: function NETGEN( $\Omega, h, X$ ):
2:  $X = \text{PNP}(\Omega, h, X_0)$  ▷ Slak and Kosec node generation.
3: Compute  $\mathcal{F}_0$  from  $X_0$  and  $\mathcal{T}$  from  $X \setminus X_0$  ▷ Geometry interfaces and cell definitions.
4:  $(r_K)_{K \in \mathcal{T}} = \text{RAD}(\mathcal{P}_{\mathcal{T}}, h, \alpha, P)$  ▷ Define cells radius.
5:  $\mathcal{F}_1 = \text{INTERF}(\mathcal{P}_{\mathcal{T}}, (r_K)_{K \in \mathcal{T}})$  ▷ Compute interior interfaces.
6:  $\mathcal{F} = \mathcal{F}_0 \cup \mathcal{F}_1$  ▷ Set interface list.
7:  $\mathcal{N} = \{\mathcal{T}, \mathcal{F}\}$  ▷ Set network.
8: if  $e$  then
9:    $\mathcal{N} = \text{ENRICH}(\mathcal{N}, n_E)$  ▷ Optional interface enrichment.
10: end if
11: return  $\mathcal{N}$ 
12: end function
```

4. Fast network geometry generation procedures

In this section we present new geometry computation procedures based on the solution of linear systems that are a simplification of the original non-linear problem of [32]. In particular, we explain that those system always have a solution and that they provide an admissible geometry if a parameter is chosen large enough. The proofs of the

various results despite their importance are lengthy, technical and in fact basic linear algebra, thus we have chosen to detail them in appendix.

4.1. The original quadratic minimization problem

In [32], the most general and robust geometry generation procedure is to solve an optimization problem of the form:

$$\mathcal{G}^* = \arg \min_{\mathcal{G} \in \mathcal{A}_{geom}^+} \mathcal{J}(\mathcal{G}),$$

where \mathcal{A}_{geom}^+ is the set of admissible geometries that satisfy equations (1), (2), (3) and (4) along with the positivity constraint on the measures and an additional constraint $m_K \leq \tau |B_K|$ for some fixed parameter $\tau > 1$. Because of the inequality constraints in \mathcal{A}_{geom}^+ , the Karush-Kuhn-Tucker first order optimality conditions associated with the minimization problem will involve the complementarity conditions

$$\Upsilon_{comp}(m_K, \lambda_K^{min}) = 0 \quad \text{and} \quad \Upsilon_{comp}(\tau |B_K| - m_K, \lambda_K^{max}) = 0,$$

where Υ_{comp} can be any complementarity function. Those additional complementarity conditions are the only non-linearity in the resulting KKT system but nevertheless force to use a non-linear solver such as Newton-Raphson's method, highly increasing the computational cost. To obtain a linear system, following closely related approaches of the literature it was proposed in [32] to rely on an heuristic choice of admissible measures and then solve the now independent d linear systems corresponding to each component of the $\eta_{K,\sigma}$'s. This gave relatively good results on not too distorted networks, however it seems difficult to provide good heuristics for challenging, distorted networks: measures seem to have a too strong impact on the geometry quality parameters to allow to completely avoid some coupling between the m_K 's and the $\eta_{K,\sigma}$'s in the optimization process (in particular, consider the example of networks based on Kershaw meshes in [32]).

4.2. A simple but surprisingly efficient linear work around

To obtain a linear system, here we propose a very basic and somewhat surprising approach. From the remark on the origin of the non-linearity of the system, a very straightforward way to reduce the computational cost is to replace the admissible set \mathcal{A}_{geom}^+ by the simpler \mathcal{A}_{geom} where we simply remove the inequality constraints:

$$\begin{aligned} \mathcal{A}_{geom} = \{ & \mathcal{G} \text{ such that } \sum_{K \in \mathcal{T}} m_K = |\Omega|, \text{ and } \sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i = m_K \varepsilon_K^{0,i} \quad \text{for all } K \in \mathcal{T} \text{ and all } 1 \leq i \leq d, \\ & \text{and } \sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i (x_\sigma^j - x_K^j) = m_K (\delta_{ij} + \varepsilon_K^{1,ij}) \quad \text{for all } K \in \mathcal{T} \text{ and all } 1 \leq i, j \leq d, \\ & \text{and } \sum_{K \in \mathcal{T}_\sigma} \eta_{K\sigma}^i = \varepsilon_\sigma^i \quad \text{for all } \sigma \in \mathcal{F}_{int} \text{ and all } 1 \leq i \leq d \}. \end{aligned}$$

Then, we consider the minimization problem:

$$\mathcal{G}^* = \arg \min_{\mathcal{G} \in \mathcal{A}_{geom}} \mathcal{J}(\mathcal{G}). \quad (8)$$

Still following [32], the most simple and computationally efficient cost function \mathcal{J} is given by:

$$\mathcal{J}(\mathcal{G}) = \frac{\delta_{\mathcal{J}}^m}{2} \sum_{K \in \mathcal{T}} m_K^2 + \frac{\delta_{\mathcal{J}}^\eta}{2} \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} |\eta_{K,\sigma}|^2 + \frac{1}{2} \sum_{K \in \mathcal{T}} \omega_{\mathcal{J},K}^0 |\varepsilon_{m,K}^0|^2 + \frac{1}{2} \sum_{K \in \mathcal{T}} \sum_{i=1}^d \sum_{j=1}^d \omega_{\mathcal{J},K}^1 |\varepsilon_{m,K}^{1,ij}|^2 + \frac{1}{2} \sum_{\sigma \in \mathcal{F}_{int}} \omega_{\mathcal{J},\sigma} |\varepsilon_\sigma|^2, \quad (9)$$

where $\delta_{\mathcal{J}}^m > 0$ and $\delta_{\mathcal{J}}^\eta > 0$ are parameters to be chosen later and where we have used the change of variable $\varepsilon_{m,K}^0 = m_K \varepsilon_K^0$ and $\varepsilon_{m,K}^{1,ij} = m_K \varepsilon_K^{1,ij}$. Still following [32], the weights $\omega_{\mathcal{J}}$ involved in formula (9) are chosen as follows:

$$\omega_{\mathcal{J},K}^0 = \frac{1}{h_K^{v+1}} \quad \omega_{\mathcal{J},K}^1 = \frac{1}{h_K^{v+2}} \quad \omega_{\mathcal{J},\sigma} = \frac{1}{\min_{K \in \mathcal{T}_\sigma} h_K^{v+1}},$$

where $\nu > 0$ is not necessarily equal to p . The Lagrangian associated with the minimization problem (8) is given by

$$\begin{aligned} \mathcal{L}(\mathcal{G}, \lambda) = & \mathcal{J}(\mathcal{G}) + \sum_{K \in \mathcal{T}} \sum_{i=1}^d \left(\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i - \varepsilon_{m,K}^{0,i} \right) \lambda_K^{0,i} + \sum_{K \in \mathcal{T}} \sum_{i=1}^d \sum_{j=1}^d \left(\eta_{K,\sigma}^i (x_\sigma^j - x_K^j) - m_K \delta_{ij} - \varepsilon_{m,K}^{1,ij} \right) \lambda_K^{1,ij} \\ & + \sum_{\sigma \in \mathcal{F}_{int}} \left(\sum_{K \in \mathcal{T}_\sigma} \eta_{K,\sigma}^i - \varepsilon_\sigma^i \right) \lambda_\sigma^i + \left(\sum_{K \in \mathcal{T}} m_K - |\Omega| \right) \lambda_\Omega, \end{aligned}$$

where:

$$\lambda = \left((\lambda_K)_{K \in \mathcal{T}}, (\lambda_K^{0,i})_{K \in \mathcal{T}, 1 \leq i \leq d}, (\lambda_K^{1,ij})_{K \in \mathcal{T}, 1 \leq i, j \leq d}, (\lambda_\sigma^i)_{\sigma \in \mathcal{F}_{int}, 1 \leq i \leq d}, \lambda_\Omega \right)$$

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

$$\partial_{\varepsilon_\sigma^i} \mathcal{L}(\mathcal{G}, \lambda) = \omega_{\mathcal{J},\sigma} \varepsilon_\sigma^i - \lambda_\sigma^i = 0 \quad \text{for all } \sigma \in \mathcal{F}_{int}, \quad 1 \leq i \leq d, \quad (10)$$

$$\partial_{\varepsilon_K^{0,i}} \mathcal{L}(\mathcal{G}, \lambda) = \omega_{\mathcal{J},K}^0 \varepsilon_{m,K}^{0,i} - \lambda_K^{0,i} = 0 \quad \text{for all } K \in \mathcal{T}, \quad 1 \leq i \leq d, \quad (11)$$

$$\partial_{\varepsilon_K^{1,ij}} \mathcal{L}(\mathcal{G}, \lambda) = \omega_{\mathcal{J},K}^1 \varepsilon_{m,K}^{1,ij} - \lambda_K^{1,ij} = 0 \quad \text{for all } K \in \mathcal{T}, \quad 1 \leq i, j \leq d, \quad (12)$$

$$\partial_{\eta_{K,\sigma}^i} \mathcal{L}(\mathcal{G}, \lambda) = \delta_{\mathcal{J}}^i \eta_{K,\sigma}^i + \sum_{j=1}^d \lambda_K^{1,ij} (x_\sigma^j - x_K^j) + \lambda_K^{0,i} + \lambda_\sigma^i = 0 \quad \text{for all } K \in \mathcal{T}, \quad \sigma \in \mathcal{F}_K, \quad 1 \leq i \leq d, \quad (13)$$

$$\partial_{m_K} \mathcal{L}(\mathcal{G}, \lambda) = \delta_{\mathcal{J}}^m m_K - \sum_{i=1}^d \lambda_K^{1,ii} + \lambda_\Omega = 0 \quad \text{for all } K \in \mathcal{T}, \quad (14)$$

complemented by the constraints:

$$\sum_{K \in \mathcal{T}} m_K = |\Omega|, \quad (1)$$

$$\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i = \varepsilon_{m,K}^{0,i} \quad \text{for all } K \in \mathcal{T} \quad \text{and all } 1 \leq i \leq d, \quad (2)$$

$$\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i (x_\sigma^j - x_K^j) = m_K \delta_{ij} + \varepsilon_{m,K}^{1,ij} \quad \text{for all } K \in \mathcal{T} \quad \text{and all } 1 \leq i, j \leq d, \quad (3)$$

$$\sum_{K \in \mathcal{T}_\sigma} \eta_{K,\sigma}^i = \varepsilon_\sigma^i \quad \text{for all } \sigma \in \mathcal{F}_{int} \quad \text{and all } 1 \leq i \leq d. \quad (4)$$

Equations (1), (2), (3), (4), (10), (11), (12), (13), (14) now simply form a linear system, that we will denote:

$$\mathbb{A}_{\mathcal{J}} \mathbf{g} = \mathbf{b} \quad \text{with} \quad \mathbf{g} = (\mathcal{G}, \lambda) \quad \text{and} \quad \mathbf{b} = (0, \dots, 0, |\Omega|)^T \quad (15)$$

This system has a very special structure that allows to eliminate most unknowns through Schur's complement, as was already noticed in [32]. Indeed, first notice that we can express the errors $\varepsilon_{m,K}^{0,i}$ from zero order consistency equations (2) and the errors $\varepsilon_{m,K}^{1,ij}$ from the first order consistency equations (3). The conservativity errors ε_σ^i are obtained from (10). In the same way, the $\eta_{K,\sigma}^i$'s and the m_K 's are obtained from respectively (13) and (14). Now, to end our Schur complement, we position the Lagrange multipliers $\lambda_K^{0,i}$ in front of equations (11) and the Lagrange multipliers $\lambda_K^{1,ij}$ in front of equations (12). All those secondary equations and secondary unknowns form our Schur complement, this choice of equations and unknowns positioning is made to ensure that the diagonal of this complement has always non-zero entries. Our final linear system then simply consists in the primary equation (1) in front of which we position the Lagrange multiplier λ_Ω and primary equations (4) in front of which we position λ_σ^i . The final linear system that actually needs to be inverted thus has only $d \text{ card}(\mathcal{F}_{int}) + 1$ equations. We gather those considerations in table 1, where the secondary equations and unknowns represent Schur's complement and the algebraic positioning of unknowns in

Table 1: Equations and unknowns positioning for Schur's complement of the linear system

	Equation	Unknown
Primary system	$\sum_{K \in \mathcal{T}_\sigma} \eta_{K\sigma}^i = \varepsilon_\sigma^i$	λ_σ^i
	$\sum_{K \in \mathcal{T}} m_K = \Omega $	λ_Ω
Secondary system	$\omega_{\mathcal{J},\sigma} \varepsilon_\sigma^i - \lambda_\sigma^i = 0$	ε_σ^i
	$\omega_{\mathcal{J},K}^0 \varepsilon_{m,K}^{0,i} - \lambda_K^{0,i} = 0$	$\lambda_K^{0,i}$
	$\omega_{\mathcal{J},K}^1 \varepsilon_{m,K}^{1,ij} - \lambda_K^{1,ij} = 0$	$\lambda_K^{1,ij}$
	$\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i = \varepsilon_{m,K}^{0,i}$	$\varepsilon_{m,K}^{0,i}$
	$\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i (x_\sigma^j - x_K^j) = m_K \delta_{ij} + \varepsilon_{m,K}^{1,ij}$	$\varepsilon_{m,K}^{1,ij}$
	$\delta_{\mathcal{J}}^i \eta_{K,\sigma}^i + \sum_{j=1}^d \lambda_K^{1,ij} (x_\sigma^j - x_K^j) + \lambda_K^{0,i} + \lambda_\sigma^i = 0$	$\eta_{K,\sigma}^i$
	$\delta_{\mathcal{J}}^m m_K - \sum_{i=1}^d \lambda_K^{1,ii} + \lambda_\Omega = 0$	m_K

front of equations, and the primary equations and unknowns are what remains in the linear system after elimination of the secondary system.

To take into account the unavoidable solver error, 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 (2), (3) and (4) applied to \mathcal{G}^* . In other words, we set

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

and

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

and

$$\bar{\varepsilon}_\sigma^i = \sum_{K \in \mathcal{T}_\sigma} \eta_{K\sigma}^{i,*} \quad \forall \sigma \in \mathcal{F}_{int}, \forall 1 \leq i \leq d$$

This allows to take into account the solver error accurately. Mirroring the existence result of admissible network geometries of [32], the following lemma ensures that a solution to (15) always exists (see Appendix A for a detailed proof).

Lemma 4.1. *The matrix $\mathbb{A}_{\mathcal{J}}$ of (15) is symmetric and invertible.*

Surprisingly enough, the unique solution to the linear problem (15) generally satisfies in practice the positivity constraint on the measures: there is thus no need to incorporate those constraints and we can simply solve a linear problem. Our understanding of this phenomenon is the following: provided $\delta_{\mathcal{J}}^m$ is large enough, at convergence the main contribution to the cost function will be the L^2 norm of the measures. Meanwhile if there exist two families of measures, one will all positive measures and one with some negative ones, then the family with positive measures will have a lower L^2 norm and therefore will most probably be selected by the minimization process. If solving this

“unconstrained” problem gives a solution with negative measures, then $\delta_{\mathcal{J}}^m$ is probably too small (or the network is very badly designed). A more rigorous analysis of this phenomenon is given by the following lemma (see Appendix B for a detailed proof):

Lemma 4.2. *Let $\delta_{\mathcal{J}}^\eta$, the $\omega_{\mathcal{J},\sigma}$'s, the $\omega_{\mathcal{J},K}^0$'s and the $\omega_{\mathcal{J},K}^1$'s be fixed to some strictly positive value. For any admissible network \mathcal{N} , there exists $\delta_{\mathcal{J}}^{m,*}(\mathcal{N})$ such that for any $\delta_{\mathcal{J}}^m > \delta_{\mathcal{J}}^{m,*}(\mathcal{N})$ the unique solution of (15) satisfies $m_K > 0$ for all $K \in \mathcal{T}$.*

From numerical experiments, our best generic choice for the weights $\delta_{\mathcal{J}}^m$ and $\delta_{\mathcal{J}}^\eta$ is $\delta_{\mathcal{J}}^m = 0.1$ and $\delta_{\mathcal{J}}^\eta = 0.01$, for $\nu = 2$.

4.3. Exact geometries

If the network satisfies one of the sufficient conditions proposed in the first part of the paper, then we know that it is legitimate to look for exact geometries. We consider to this end the set:

$$\begin{aligned} \mathcal{A}_{geom}^{ex} = & \left\{ \mathcal{G}_r \text{ such that } \sum_{K \in \mathcal{T}} m_K = |\Omega|, \text{ and } \sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i = 0 \text{ for all } K \in \mathcal{T} \text{ and all } 1 \leq i \leq d, \right. \\ & \text{and } \sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i (x_\sigma^j - x_K^j) = m_K \delta_{ij} \text{ for all } K \in \mathcal{T} \text{ and all } 1 \leq i, j \leq d, \\ & \left. \text{and } \sum_{K \in \mathcal{T}_\sigma} \eta_{K,\sigma}^i = 0 \text{ for all } \sigma \in \mathcal{F}_{int} \text{ and all } 1 \leq i \leq d \right\}. \end{aligned}$$

For elements of \mathcal{A}_{geom}^{ex} , the cost function \mathcal{J} simplifies into \mathcal{J}_r :

$$\mathcal{J}_r(\mathcal{G}_r) = \frac{\delta_{\mathcal{J}}^m}{2} \sum_{K \in \mathcal{T}} m_K^2 + \frac{\delta_{\mathcal{J}}^\eta}{2} \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} |\eta_{K,\sigma}|^2,$$

and the Lagrangian associated with the minimization problem:

$$\mathcal{G}_r^* = \arg \min_{\mathcal{G}_r \in \mathcal{A}_{geom}^{ex}} \mathcal{J}_r(\mathcal{G}_r), \quad (16)$$

is equivalent to getting rid of the now useless approximation errors:

$$\begin{aligned} \mathcal{L}_r(\mathcal{G}_r, \lambda_r) = & \mathcal{J}_r(\mathcal{G}_r) + \sum_{K \in \mathcal{T}} \sum_{i=1}^d \left(\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i \right) \lambda_K^{0,i} + \sum_{K \in \mathcal{T}} \sum_{i=1}^d \sum_{j=1}^d \left(\eta_{K,\sigma}^i (x_\sigma^j - x_K^j) - m_K \delta_{ij} \right) \lambda_K^{1,ij} \\ & + \sum_{\sigma \in \mathcal{F}_{int}} \left(\sum_{K \in \mathcal{T}_\sigma} \eta_{K,\sigma}^i \right) \lambda_\sigma^i + \left(\sum_{K \in \mathcal{T}} m_K - |\Omega| \right) \lambda_\Omega. \end{aligned}$$

The Karush-Kuhn-Tucker first order optimality conditions associated with this reduced system are given by (setting again $\lambda_\sigma^i = 0$ for any $\sigma \in \mathcal{F}_{ext}$ and any $1 \leq i \leq d$):

$$\partial_{\eta_{K,\sigma}^i} \mathcal{L}_r(\mathcal{G}_r, \lambda_r) = \delta_{\mathcal{J}}^\eta \eta_{K,\sigma}^i + \sum_{j=1}^d \lambda_K^{1,ij} (x_\sigma^j - x_K^j) + \lambda_K^{0,i} + \lambda_\sigma^i = 0, \quad (17)$$

$$\partial_{m_K} \mathcal{L}_r(\mathcal{G}_r, \lambda_r) = \delta_{\mathcal{J}}^m m_K - \sum_{i=1}^d \lambda_K^{1,ii} + \lambda_\Omega = 0, \quad (18)$$

complemented by the exact constraints:

$$\sum_{K \in \mathcal{T}} m_K = |\Omega|, \quad (1)$$

$$\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i = 0 \quad \text{for all } K \in \mathcal{T} \text{ and all } 1 \leq i \leq d, \quad (19)$$

$$\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i (x_\sigma^j - x_K^j) = m_K \delta_{ij} \quad \text{for all } K \in \mathcal{T} \text{ and all } 1 \leq i, j \leq d, \quad (20)$$

$$\sum_{K \in \mathcal{T}_\sigma} \eta_{K,\sigma}^i = 0 \quad \text{for all } \sigma \in \mathcal{F}_{int} \text{ and all } 1 \leq i \leq d. \quad (21)$$

Equations (1), (19), (20), (21), (17), (18) again simply form a linear system, that we will denote:

$$\mathbb{A}_{\mathcal{J},r} \mathbf{g}_r = \mathbf{b}_r \quad \text{with} \quad \mathbf{g}_r = (\mathcal{G}_r, \lambda_r) \quad \text{and} \quad \mathbf{b}_r = (0, \dots, 0, |\Omega|)^T \quad (22)$$

The same kind of Schur complement can be performed on this reduced system: the $\eta_{K,\sigma}^i$'s and the m_K 's are still obtained from respectively (17) and (18). However, this time to build our Schur complement we position the Lagrange multipliers $\lambda_K^{0,i}$ in front of equations (19) and the Lagrange multipliers $\lambda_K^{1,ij}$ in front of equations (20). The final linear system still consists in equation (1) in front of which we position the Lagrange multiplier λ_Ω and equations (21) in front of which we position λ_σ^i , leading again to a linear system of size $d \text{ card}(\mathcal{F}_{int}) + 1$ (see table 2). Thus, from the purely algebraic perspective, enforcing exactness of the geometry simply reduces the cost of performing Schur's complement, while potentially deteriorating the condition number.

Table 2: Equations and unknowns positioning for Schur's complement of the linear system for exact geometries

	Equation	Unknown
Primary system	$\sum_{K \in \mathcal{T}_\sigma} \eta_{K,\sigma}^i = 0$	λ_σ^i
	$\sum_{K \in \mathcal{T}} m_K = \Omega $	λ_Ω
Secondary system	$\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i = 0$	$\lambda_K^{0,i}$
	$\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i (x_\sigma^j - x_K^j) = m_K \delta_{ij}$	$\lambda_K^{1,ij}$
	$\delta_{\mathcal{J}}^n \eta_{K,\sigma}^i + \sum_{j=1}^d \lambda_K^{1,ij} (x_\sigma^j - x_K^j) + \lambda_K^{0,i} + \lambda_\sigma^i = 0$	$\eta_{K,\sigma}^i$
	$\delta_{\mathcal{J}}^m m_K - \sum_{i=1}^d \lambda_K^{1,ii} + \lambda_\Omega = 0$	m_K

Existence of a solution to (22) is ensured by the following lemma (see Appendix C):

Lemma 4.3. *The matrix $\mathbb{A}_{\mathcal{J},r}$ of (22) is symmetric and there always exists a solution to the critical point linear problem (22) associated with (16). Moreover, if the sufficient condition (H) for existence of exact geometries is satisfied, then $\text{Ker } \mathbb{A}_{\mathcal{J},r} = \{0\}$.*

If a solution to (22) always exists, to be a true exact geometry once again we need the $(m_K)_{K \in \mathcal{T}}$ to be positive. Mirroring the case of admissible geometries, it is easy to establish that (see Appendix D):

Lemma 4.4. *Let $\delta_{\mathcal{J}}^n$, the $\omega_{\mathcal{J},\sigma}$'s, the $\omega_{\mathcal{J},K}^0$'s and the $\omega_{\mathcal{J},K}^1$'s be fixed to some strictly positive value. If the sufficient condition (H) for existence of exact geometries is satisfied, then there exists $\delta_{\mathcal{J}}^{m,*}(\mathcal{N})$ such that for any $\delta_{\mathcal{J}}^m > \delta_{\mathcal{J}}^{m,*}(\mathcal{N})$ the unique critical point associated with (16) satisfies $m_K > 0$ for all $K \in \mathcal{T}$.*

4.4. Exactly compatible geometries

In the intermediate case of exactly compatible admissible geometries, one simply solves the system presented in table 3, which is of course the system for admissible geometries without the error terms ε_σ^i for compatibility. An easy adaptation of the proofs of lemma 4.1 and 4.2 for the case of an admissible geometry gives the following result:

Corollary 4.5. *The matrix $\mathbb{A}_{\mathcal{J},c}$ underlying the system of table 3 is symmetric and invertible. Let $\delta_{\mathcal{J}}^\eta$, the $\omega_{\mathcal{J},\sigma}$'s, the $\omega_{\mathcal{J},K}^0$'s and the $\omega_{\mathcal{J},K}^1$'s be fixed to some strictly positive value. For any admissible network \mathcal{N} , there exists $\delta_{\mathcal{J}}^{m,*}(\mathcal{N})$ such that for any $\delta_{\mathcal{J}}^m > \delta_{\mathcal{J}}^{m,*}(\mathcal{N})$ the unique solution to the linear system of table 3 satisfies $m_K > 0$ for all $K \in \mathcal{T}$.*

Table 3: Equations and unknowns positioning for Schur's complement of the linear system for exactly compatible geometries

	Equation	Unknown
Primary system	$\sum_{K \in \mathcal{T}_\sigma} \eta_{K\sigma}^i = 0$	λ_σ^i
	$\sum_{K \in \mathcal{T}} m_K = \Omega $	λ_Ω
Secondary system	$\omega_{\mathcal{J},K}^0 \varepsilon_{m,K}^{0,i} - \lambda_K^{0,i} = 0$	$\lambda_K^{0,i}$
	$\omega_{\mathcal{J},K}^1 \varepsilon_{m,K}^{1,ij} - \lambda_K^{1,ij} = 0$	$\lambda_K^{1,ij}$
	$\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i = \varepsilon_{m,K}^{0,i}$	$\varepsilon_{m,K}^{0,i}$
	$\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i (x_\sigma^j - x_K^j) = m_K \delta_{ij} + \varepsilon_{m,K}^{1,ij}$	$\varepsilon_{m,K}^{1,ij}$
	$\delta_{\mathcal{J}}^\eta \eta_{K,\sigma}^i + \sum_{j=1}^d \lambda_K^{1,ij} (x_\sigma^j - x_K^j) + \lambda_K^{0,i} + \lambda_\sigma^i = 0$	$\eta_{K,\sigma}^i$
	$\delta_{\mathcal{J}}^m m_K - \sum_{i=1}^d \lambda_K^{1,ii} + \lambda_\Omega = 0$	m_K

5. Numerical exploration

Let us begin this numerical section by mentioning that thanks to the symmetry of the system, we found that the conjugate gradient method (CG) combined with symmetric successive over-relaxation (SSOR) as preconditioner offers a robust choice for the inversion of the linear system underlying geometry computation. Of course in the following this choice of solver and preconditioner is used for every simulation.

To assess the quality and robustness of the geometry generation, we will consider in all the following the network element method of [32] applied to the Poisson problem $-\Delta u = f$ for the analytic solution $u(\mathbf{x}) = \prod_{i=1}^d \sin(\pi x^i)$ with Dirichlet boundary conditions.

In dimension 2, the parameter for candidates generation in the algorithm 1 of Slak and Kosec is set to $k_0 = 15$, while in dimension 3 we use $k_0 = 6$ and $p_0 = 6$. We also set $P = d + 1$ in the interface generation algorithm 3. Finally, when we seek exact geometries we use the interface enrichment procedure with $n_e = d + 1$ to ensure that (H1) is satisfied.

5.1. Constant size functions

For the 2d square domain $\Omega =]0, 1]^2$, applying algorithm 4 with a constant size function produces the cell nodes and interfaces of figure 1. We also consider a distorted version of this network, applying the distortion introduced in

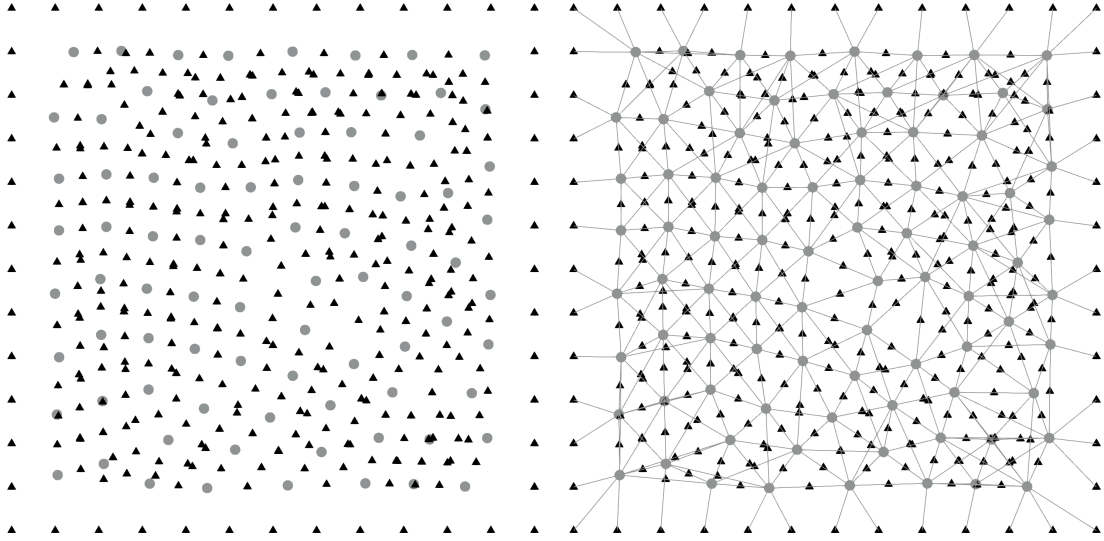


Figure 1: Cell nodes in dimension 2 for the square domain, triangles are interfaces, circles are cell nodes. Left without connectivity, right with connectivity

[31] for meshes to both our cell and interfaces nodes (except boundary interfaces):

$$\begin{aligned} x' &= x + 7(1-x-y)xy(1-x)(1-y), \\ y' &= y + 7(1-x-y)xy(1-x)(1-y). \end{aligned} \quad (23)$$

This produces the distorted networks of figure 2. In dimension 3, we consider the unit ball $\Omega = B(0, 1)$. The cell

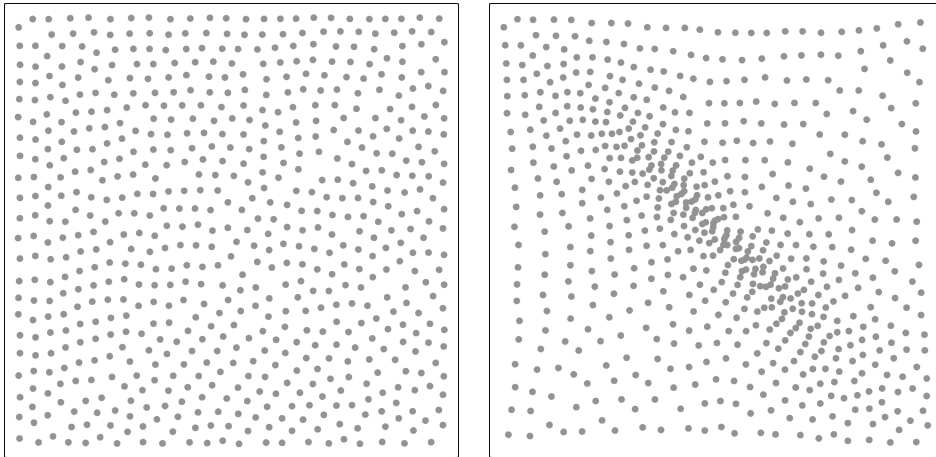


Figure 2: Cell nodes in dimension 2 : left reference cloud, right distorted cloud

nodes generated by algorithm 4 with constant size function are displayed on figure 3. On those networks, we solve both the admissible (15) and exact (22) geometry problems, and then use the obtained geometries for the NEM Poisson problem: convergence curves are displayed on figure 4, while approximate convergence orders are presented in table 4.

The results of figure 4 and table 4 confirms that the generated networks and geometry are of good quality: we indeed obtain the optimal second order convergence that was guaranteed in [32] only by the non-linear geometry computation.

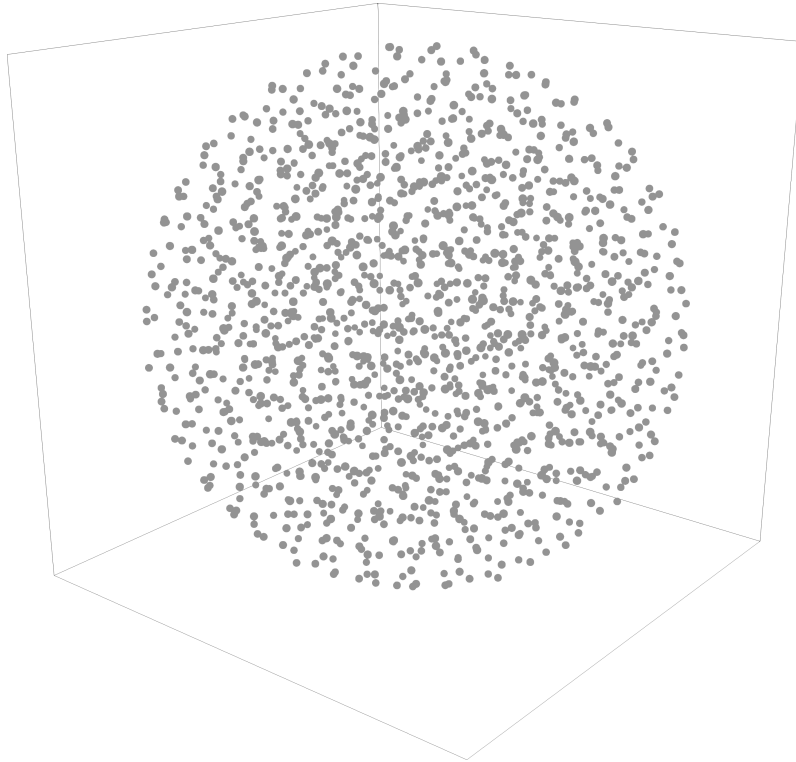


Figure 3: Sphere cell nodes

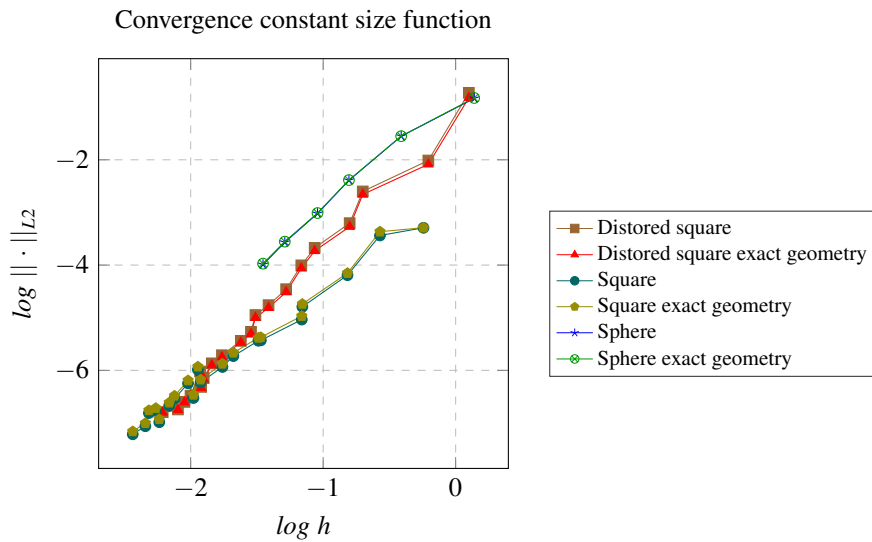


Figure 4: Convergence curves for the networks generated by algorithm 4 with constant size function

This confirms that our very simple linear work around consisting in dropping the inequality constraints is a working solution. Notice that in those cases, no true difference in solution precision is observed between admissible and exact geometries.

Table 4: Approximate orders of convergence

	Admissible geometry	Exact geometry
Square	1.88	1.89
Square-distorted	2.69	2.63
Sphere	2.01	2.01

5.2. Variable size functions



Figure 5: Models for variable size functions : dog and rooster (photo credits : Michael Dzedzic and Sahand Babali)



Figure 6: Models for variable size functions : owl and white bear (photo credits : Luis Argaiz and Hans Jurgen Mager)

To generate variable size functions with uncontrolled size distribution we consider the animal pictures displayed

in figures 5 and 6. All those pictures have been downloaded from the Unsplash free repository (<https://unsplash.com>). We rework those pictures by first reducing the resolution so that they are all approximately 1000 pixels large in their largest direction. Then, we transform them into gray scale images (.pgm format) and finally we use them to define a size function on a rectangle with largest size equal to one and the same side length ratio as the rectangular images they are defined from. The size function on the rectangular domain then follows the gray scale of the image, darker gray meaning a smaller value for the local size, and the networks are generated from algorithm 4.

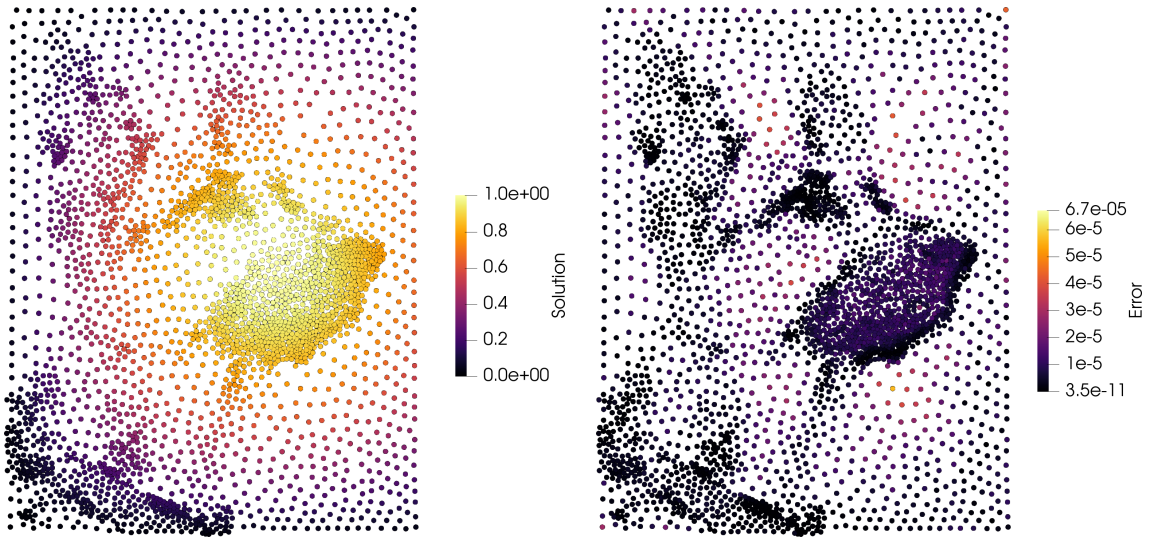


Figure 7: Dog size function

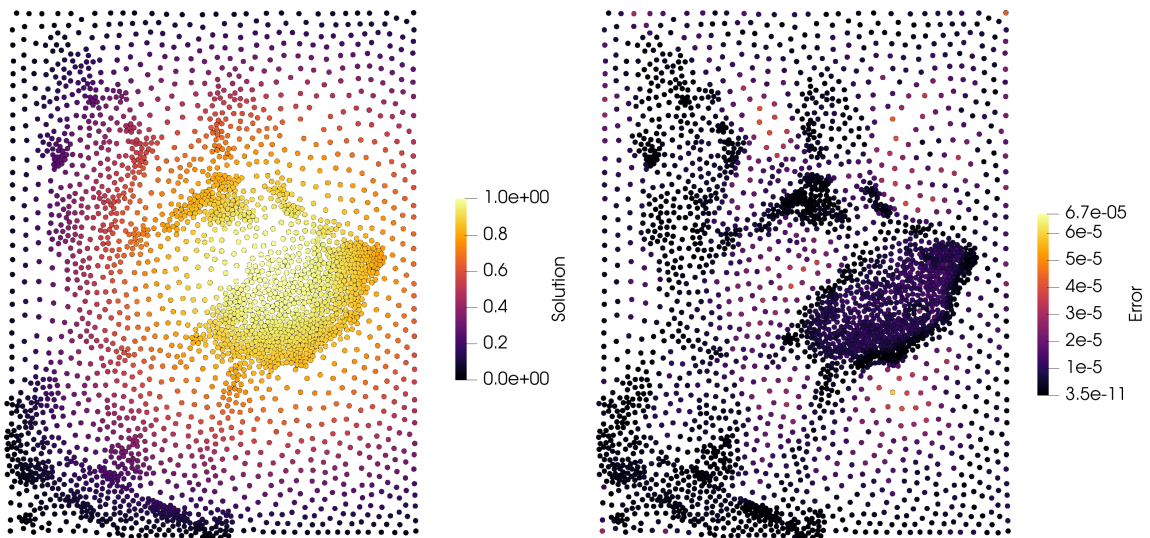


Figure 8: Dog size function : exact geometry

We display the solution values and error values for each image, both for admissible and exact geometries on figures 7, 8, 9, 10, 11, 12, 13 and 14, also revealing this way the cell nodes distributions obtained from the irregular size functions. In table 5, we gather the L^2 norm of the error for each configuration.

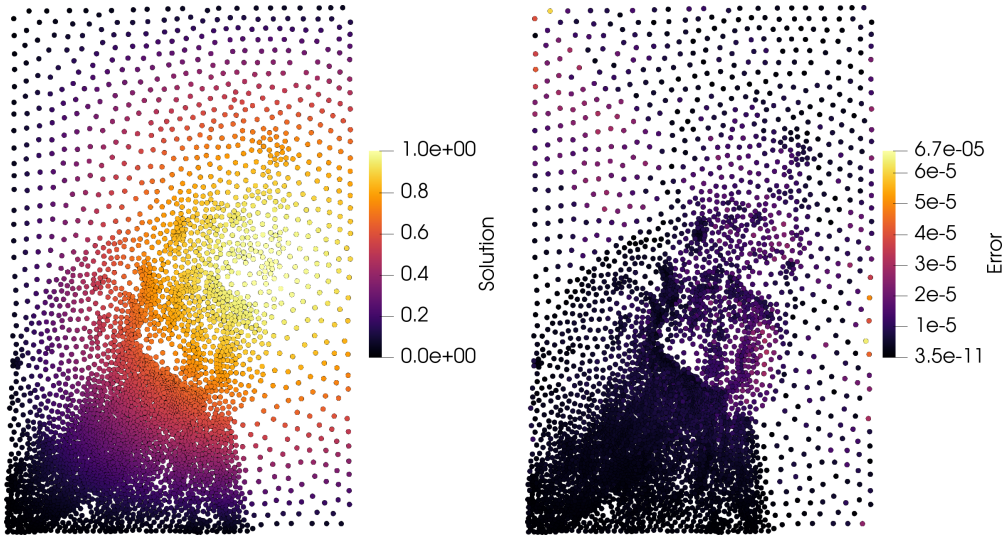


Figure 9: Rooster size function

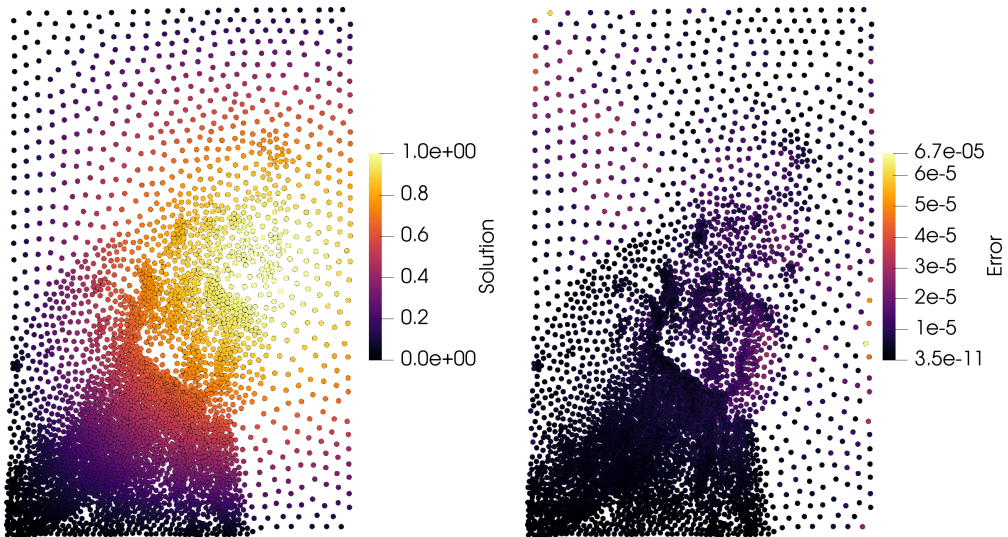


Figure 10: Rooster size function : exact geometry

Table 5: Approximate orders of convergence

	Admissible geometry L^2 error	Exact geometry L^2 error	h
Dog	5.2111e-4	5.323e-4	0.0703
Rooster	5.120e-4	6.743e-4	0.0596
Owl	6.743e-4	6.932e-4	0.0596
White bear	1.792e-4	1.807e-4	0.0574

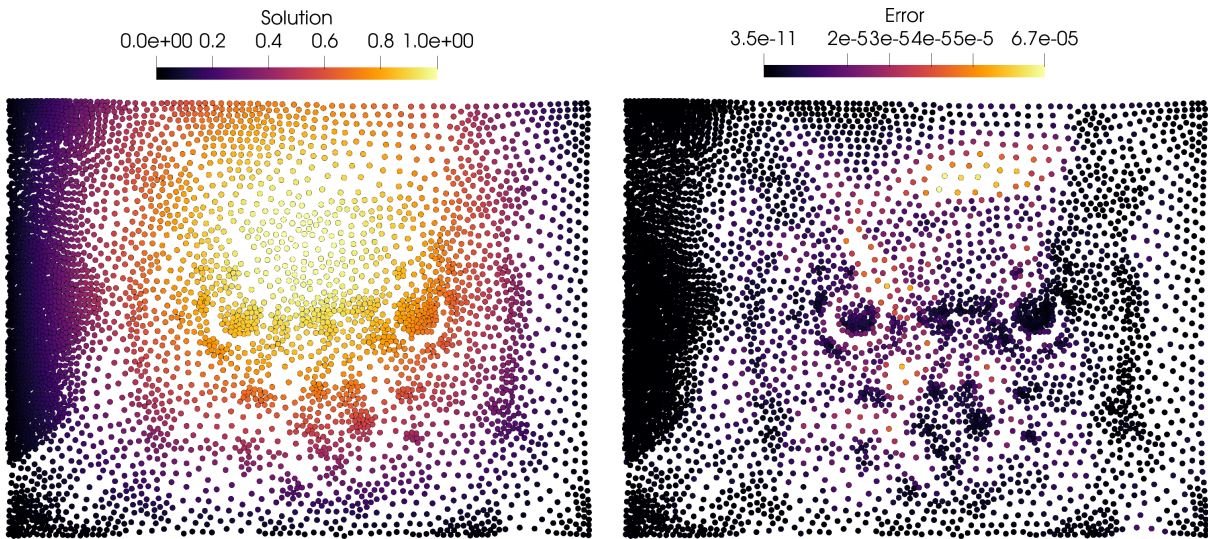


Figure 11: Owl size function

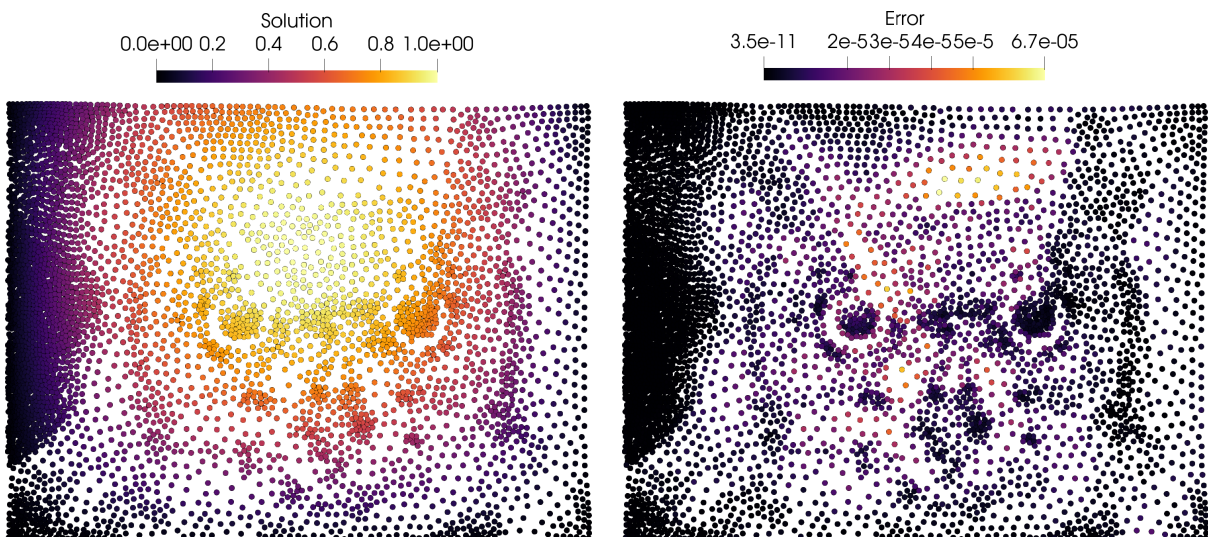


Figure 12: Owl size function : exact geometry

From the figures we see that the larger errors correspond to the larger cell sizes and that the node distribution have little impact on it, confirming the robustness of the geometry generation procedure regarding node distribution. In table 5 we see that solving the linear problems (15) and (22) for computing admissible and exact network geometries produces admissible geometries giving slightly better results than exact ones. Clearly, a better tuning of the cost function \mathcal{J}_r could improve the obtained exact geometries. This also emphasizes the fact that the freedom authorized by admissible only geometries can lead to better approximation results contradicting the intuitive idea that an exact geometry will always produce better results.

5.3. Networks from distorted meshes

In [32] to illustrate the robustness of the network element method with respect to point cloud distortion, some networks were generated from distorted meshes by using the mesh cell centers as cell nodes and the mesh vertices

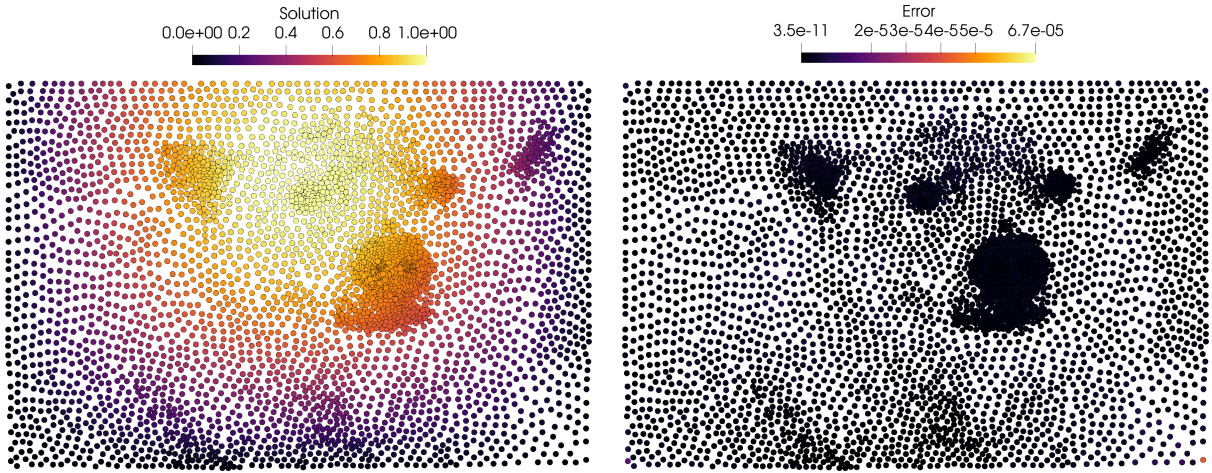


Figure 13: White bear size function

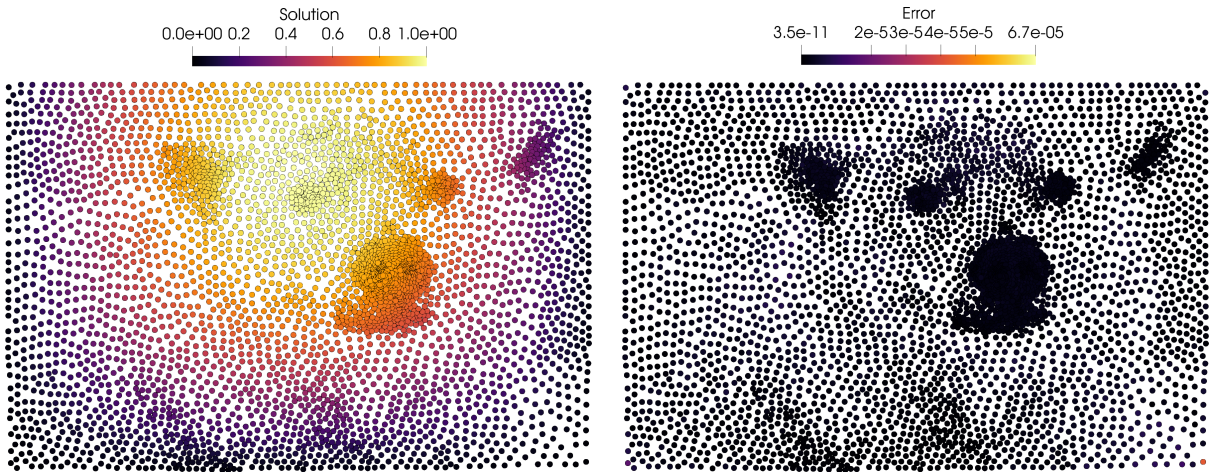


Figure 14: White bear size function : exact geometry

as interfaces. In those situations the original non-linear geometry problem of [32] produces good quality geometries, while the linear work around proposed in [32] and recalled at the beginning of the previous section that consists in prescribing heuristic values for the cell measures $(m_K)_{K \in \mathcal{T}}$ and then solving d decoupled linear systems, one for each component of the remaining elements of \mathcal{G} gave good results only for not too distorted geometries, and in general much less good than those obtained from the non-linear problem. In particular, it produces geometries with unacceptable quality for the Kershaw mesh sequence illustrated by figure 15. Defining our networks from meshes in the same way, we test our linear network geometry generation procedure on both the Kershaw mesh sequence and also on the distorted triangular meshes of [31] obtained by applying transformation 23 to a regular triangular mesh (see figure 15). As we have a background mesh, we compare the NEM results with a first order virtual element method (VEM, [6]).

Convergence curves are displayed on figure 16 while approximate convergence orders are given in table 6. We see that on the distorted triangular mesh sequences, the NEM, NEM with exact geometry and VEM produce almost identical results. On the Kershaw mesh, the two NEM give even slightly better results than the NEM, with admissible geometries performing better than exact ones as it can produce more balanced quality parameters by allowing some

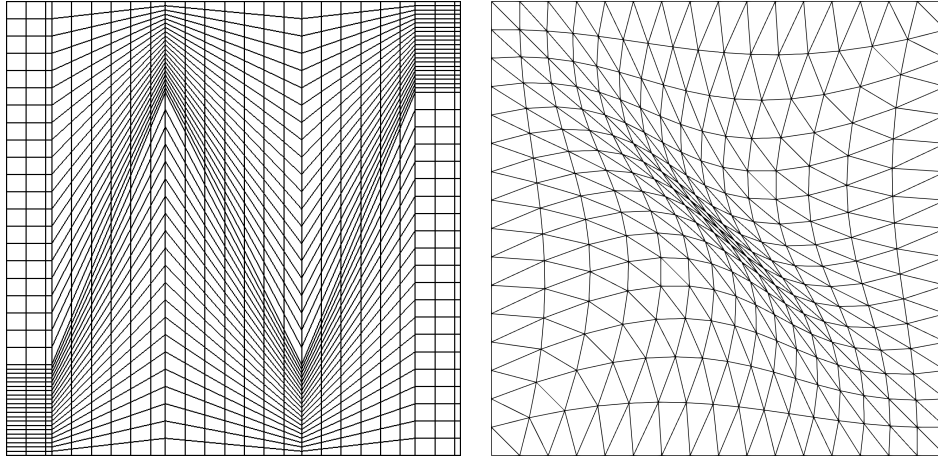


Figure 15: Example of Kershaw mesh and distorted triangular mesh

Convergence: networks from distorted meshes

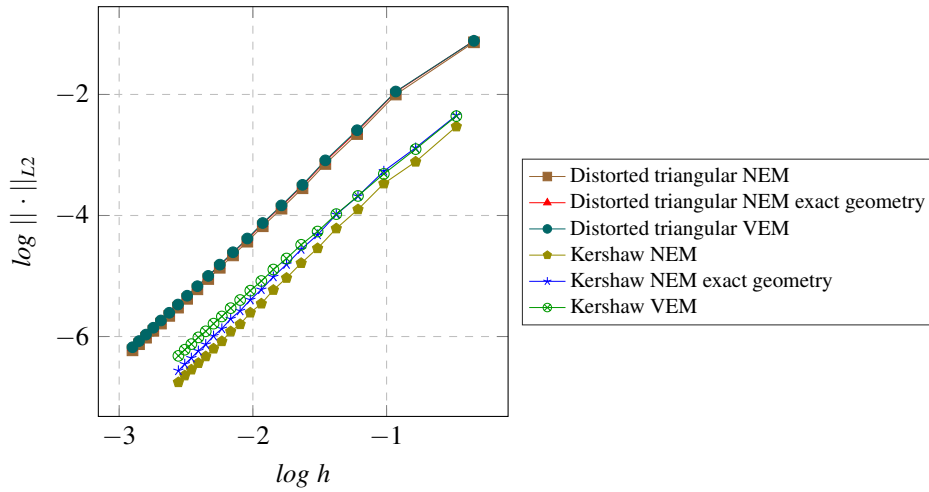


Figure 16: Convergence curves for the networks generated by algorithm 4 with constant size function

Table 6: Approximate orders of convergence

	Admissible geometry	Exact geometry	Virtual element
Kershaw	2.13	2.18	1.99
Distorted triangular	2.10	2.10	2.10

small geometric error. In any case, this reveals that our new linear work around do produce good quality geometries on those artificial but challenging configurations, for which the linear approach of [32] was failing.

6. Conclusion and perspectives

After presenting some sufficient conditions ensuring the existence of exact network geometries, we proposed a network generation algorithm that if needed allows to fulfill one of those sufficient conditions. Next, we reconsidered the non-linear problem originally used in [32] to generate good quality network geometries, and proposed to solve instead a linear system whose solution will satisfy the required positivity constraint on the cell measures provided a parameter is chosen large enough. We then explained how this could be adapted to the practical computation of exact geometries, provided one of the sufficient conditions of the first part is fulfilled. Finally, we concluded by some numerical experiments on network and geometry generation illustrating the robustness of both the network and geometry generation procedures. It is our belief that the results of the present paper now make the overall workflow of the network element method, from network generation to the actual computation of a solution, fast enough for the NEM to have some true practical interest as a meshfree method. Of course, there is still much room for improvement: in particular, it is obvious that the proposed network generation algorithm produces probably too many interfaces and therefore more advanced binning algorithms could produce well connected networks with less interfaces. In the same way, at the very least the weights $\delta_{\mathcal{J}}^m$ and $\delta_{\mathcal{J}}^l$ in the cost function \mathcal{J} should probably be made variable in each cell to take into account the varying cell diameter h_K and more generally the weighting of function \mathcal{J} could be improved.

References

- [1] D. N. Arnold, F. Brezzi, B. Cockburn, L. Marini, Unified analysis of discontinuous galerkin methods for elliptic problems, *SIAM J. Numer. Anal.*, Vol. 39(5), pp. 1749-1779 (2002).
- [2] D. A. Di Pietro, A. Ern, *Mathematical aspects of discontinuous Galerkin methods*, Springer, 2012.
- [3] F. Brezzi, K. Lipnikov, V. Simoncini, A family of mimetic finite difference methods on polygonal and polyhedral meshes, *Math. Models Methods Appl. Sci.*, Vol. 15(10), pp. 1533-1551 (2005).
- [4] F. Brezzi, K. Lipnikov, M. Shashkov, Convergence of the mimetic finite difference method for diffusion problems on polyhedral meshes, *SIAM J. Numer. Anal.*, Vol. 43(5), pp. 1872-1896 (2005).
- [5] L. Beirão da Veiga, K. Lipnikov, G. Manzini, *The Mimetic Finite Difference Method for Elliptic Problems*, Springer, 2014.
- [6] 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).
- [7] R. Eymard, T. Gallouët, R. Herbin, A new finite volume scheme for anisotropic diffusion problems on general grids: convergence analysis, *C. R., Math., Acad. Sci. Paris*, Vol. 344(6), pp. 403-406 (2007).
- [8] R. Eymard, T. Gallouët, R. Herbin, Discretisation of heterogeneous and anisotropic diffusion problems on general nonconforming meshes sush: a scheme using stabilisation and hybrid interfaces, *IMA J. Num. Anal.*, 30(4), pp 1009-1043 (2010).
- [9] 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).
- [10] 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).
- [11] L. Lucy, A numerical approach to the testing of the fission hypothesis, *Astron. J.*, Vol. 82 pp. 1013-1024 (1977).
- [12] R. Gingold, J. Monaghan, Smoothed particle hydrodynamics: theory and application to non-spherical stars, *Mon. Not. R. Astron. Soc.*, Vol. 181 pp. 375-389 (1977).
- [13] W. Liu, S. Jun, Y. Zhang, Reproducing kernel particle methods, *Int. J. Numer. Methods Fluids*, Vol. 20, pp. 1081-1106 (1995).
- [14] P. Jensen, Finite difference techniques for variable grids, *Comput. Struct.* Vol. 2, pp. 17-29 (1972).
- [15] J. M. Melenk, *On Approximation in Meshless Methods*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 65–141.
- [16] E. O. nate, S. Idelsohn, O. Zienkiewicz, R. Taylor, A finite point method in computational mechanics, applications to convective transport and fluid flow, *Int. J. Numer. Methods Eng.*, Vol.39, pp. 3839-3866 (1996).
- [17] R. Hardy, Multiquadric equations of topography and other irregular surfaces, *J. Geophys. Res.*, Vol. 76, pp. 1905 (1971).
- [18] R. Hardy, Theory and applications of the multiquadric-biharmonic method 20 years of discovery 1968– 1988, *Comput. Math. with Appl.*, Vol. 19, pp.163-208 (1990).
- [19] C. Franke, R. Schaback, Solving partial differential equations by collocation using radial basis functions, *Appl. Math. Comput.*, Vol. 93, pp. 73-82 (1998).
- [20] B. Nayroles, G. Touzot, P. Villon, Generalizing the finite element method: Diffuse approximation and diffuse elements, *Comput. Mech.* Vol. 10, pp. 307–318 (1992).
- [21] T. Belytschko, Y. Lu, L. Gu, Element-free galerkin methods, *Int. J. Numer. Methods Eng.* Vol. 37, pp. 229-256 (1994).
- [22] Y. Lu, T. Belytschko, L. Gu, A new implementation of the element free galerkin method, *Comput. Methods Appl. Mech. Eng.*, Vol. 113, pp. 397-414 (1994).
- [23] 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).
- [24] J. Melenk, I. Babuška, The partition of unity method, *Int. J. Numer. Methods Eng.*, Vol. 40, pp. 727-758 (1997).
- [25] J.-S. Chen, M. Hillman, S.-W. Chi, Meshfree methods: Progress made after 20 years, *Journal of Engineering Mechanics*, Vol. 143(4) (2017).
- [26] A. Katz, A. Jameson, A meshless volume scheme, *Proceedings of 19th AIAA Computational Fluid Dynamics, Fluid Dynamics and Collocated Conferences*, pp. 2009-3534, American Institute of Aeronautics and Astronautics (2009).

- [27] O. Diyanov, Uncertain grid method for numerical solution of pdes, Technical report, NeurOK Software (2008).
- [28] 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).
- [29] 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).
- [30] 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).
- [31] 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).
- [32] J. Coatléven, Principles of a network element method, *Journal of Computational Physics*, vol. 433, Art. 110197 (2021).
- [33] J. Coatléven, Basic convergence theory for the network element method, *ESAIM: Mathematical Modelling and Numerical Analysis*, Vol. 55(5), pp. 2503-2533 (2021).
- [34] J. Coatléven, A network element method for heterogeneous and anisotropic diffusion-reaction problems, *Journal of Computational Physics*, Vol. 470, Art. 111597 (2022).
- [35] J. Slak, G. Kosec, On generation of node distributions for meshless pde discretizations, *SIAM Journal on Scientific Computing*, Vol. 41(5), pp. A3202-A3229 (2019).
- [36] U. Duh, M. Depolli, J. Slak, G. Kosec, Parallel point sampling for 3d bodies, *Proceedings of the 43rd International Convention on Information, Communication and Electronic Technology*, (MIPRO), pp. 219-223 (2020).
- [37] K. Van der Sande, B. Fornberg, Fast variable density 3-d node generation, *SIAM Journal on Scientific Computing*, Vol. 43(1), pp. A242-A257 (2021).
- [38] B. Fornberg, N. Flyer, Fast generation of 2-d node distributions for mesh-free pde discretizations, *Computers & Mathematics with Applications*, Vol. 69, pp. 531-544 (2015).
- [39] V. Shankar, R. M. Kirby, A. L. Fogelson, Robust node generation for meshfree discretizations on irregular domains and surfaces, *SIAM J. Sci. Comput.*, Vol. 40, pp. 2584–2608 (2018).

Appendix A. Proof of lemma 4.1

We have:

$$\begin{aligned}
& (\mathbb{A}_{\mathcal{J}} \mathbf{g})^T \hat{\mathbf{g}} = \\
& \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d \left(\delta_{\mathcal{J}}^{\eta} \hat{\eta}_{K,\sigma}^i + \sum_{j=1}^d \lambda_K^{1,ij} (x_{\sigma}^j - x_K^j) + \lambda_K^{0,i} + \lambda_{\sigma}^i \right) \hat{\eta}_{K,\sigma}^i + \sum_{K \in \mathcal{T}} \left(\delta_{\mathcal{J}}^m m_K - \sum_{i=1}^d \lambda_K^{1,ii} + \lambda_{\Omega} \right) \hat{m}_K \\
& + \left(\sum_{K \in \mathcal{T}} m_K \right) \hat{\lambda}_{\Omega} + \sum_{\sigma \in \mathcal{F}_{int}} \sum_{i=1}^d \left(\sum_{K \in \mathcal{T}_{\sigma}} \eta_{K,\sigma}^i - \varepsilon_{\sigma}^i \right) \hat{\varepsilon}_{\sigma}^i \\
& + \sum_{K \in \mathcal{T}} \sum_{i=1}^d \left(\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i - \varepsilon_{m,K}^{0,i} \right) \hat{\lambda}_K^{0,i} + \sum_{K \in \mathcal{T}} \sum_{i=1}^d \sum_{j=1}^d \left(\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i (x_{\sigma}^j - x_K^j) - m_K \delta_{ij} - \varepsilon_{m,K}^{1,ij} \right) \hat{\lambda}_K^{1,ij} \\
& + \sum_{K \in \mathcal{T}} \sum_{i=1}^d \left(\omega_{\mathcal{J},K}^0 \varepsilon_{m,K}^{0,i} - \lambda_K^{0,i} \right) \hat{\varepsilon}_{m,K}^{0,i} + \sum_{K \in \mathcal{T}} \sum_{i=1}^d \sum_{j=1}^d \left(\omega_{\mathcal{J},K}^1 \varepsilon_{m,K}^{1,ij} - \lambda_K^{1,ij} \right) \hat{\varepsilon}_{m,K}^{1,ij} + \sum_{\sigma \in \mathcal{F}_{int}} \sum_{i=1}^d \left(\omega_{\mathcal{J},\sigma} \varepsilon_{\sigma}^i - \lambda_{\sigma}^i \right) \hat{\varepsilon}_{\sigma}^i,
\end{aligned}$$

leading to:

$$\begin{aligned}
\mathbf{g} \mathbb{A}_{\mathcal{J}}^T \hat{\mathbf{g}} &= \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_{int}} \sum_{i=1}^d \hat{\eta}_{K,\sigma}^i \left(\delta_{\mathcal{J}}^{\eta} \hat{\eta}_{K,\sigma}^i + \hat{\lambda}_K^{0,i} + \sum_{j=1}^d \hat{\lambda}_K^{1,ij} (x_{\sigma}^j - x_K^j) + \hat{\varepsilon}_{\sigma}^i \right) \\
& + \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K \cap \mathcal{F}_{ext}} \sum_{i=1}^d \hat{\eta}_{K,\sigma}^i \left(\delta_{\mathcal{J}}^{\eta} \hat{\eta}_{K,\sigma}^i + \hat{\lambda}_K^{0,i} + \sum_{j=1}^d \hat{\lambda}_K^{1,ij} (x_{\sigma}^j - x_K^j) \right) \\
& + \sum_{K \in \mathcal{T}} m_K \left(\delta_{\mathcal{J}}^m \hat{m}_K + \hat{\lambda}_{\Omega} - \sum_{i=1}^d \hat{\lambda}_K^{1,ii} \right) + \lambda_{\Omega} \left(\sum_{K \in \mathcal{T}} \hat{m}_K \right) + \sum_{K \in \mathcal{T}} \sum_{i=1}^d \hat{\lambda}_K^{0,i} \left(\sum_{\sigma \in \mathcal{F}_K} \hat{\eta}_{K,\sigma}^i - \varepsilon_{m,K}^{0,i} \right) \\
& + \sum_{K \in \mathcal{T}} \sum_{i=1}^d \sum_{j=1}^d \hat{\lambda}_K^{1,ij} \left(\sum_{\sigma \in \mathcal{F}_K} \hat{\eta}_{K,\sigma}^i (x_{\sigma}^j - x_K^j) - \delta_{ij} \hat{m}_K - \hat{\varepsilon}_{m,K}^{1,ij} \right) + \sum_{\sigma \in \mathcal{F}_{int}} \sum_{i=1}^d \lambda_{\sigma}^i \left(\sum_{K \in \mathcal{T}_{\sigma}} \hat{\eta}_{K,\sigma}^i - \varepsilon_{\sigma}^i \right)
\end{aligned}$$

$$+ \sum_{K \in \mathcal{T}} \sum_{i=1}^d \left(\omega_{\mathcal{J},K}^0 \hat{\varepsilon}_{m,K}^{0,i} - \hat{\lambda}_K^{0,i} \right) \varepsilon_{m,K}^{0,i} + \sum_{K \in \mathcal{T}} \sum_{i=1}^d \sum_{j=1}^d \left(\omega_{\mathcal{J},K}^1 \hat{\varepsilon}_{m,K}^{1,ij} - \hat{\lambda}_K^{1,ij} \right) \varepsilon_{m,K}^{1,ij} + \sum_{\sigma \in \mathcal{F}_{int}} \sum_{i=1}^d \left(\omega_{\mathcal{J},\sigma} \hat{\varepsilon}_{\sigma}^i - \hat{\lambda}_{\sigma}^i \right) \varepsilon_{\sigma}^i,$$

which clearly shows that $\mathbb{A}_{\mathcal{J}}^T = \mathbb{A}_{\mathcal{J}}$. Meanwhile, we have:

$$\mathbf{b}^T \hat{\mathbf{g}} = |\Omega| \hat{\lambda}_{\Omega}.$$

Assume that $\mathbb{A}_{\mathcal{J}}^T \hat{\mathbf{g}} = 0$. Immediately, we deduce that:

$$\begin{aligned} \omega_{\mathcal{J},K}^0 \hat{\varepsilon}_{m,K}^{0,i} - \hat{\lambda}_K^{0,i} &= 0 \quad \text{for all } K \in \mathcal{T}, \quad 1 \leq i \leq d, \\ \omega_{\mathcal{J},K}^1 \hat{\varepsilon}_{m,K}^{1,ij} - \hat{\lambda}_K^{1,ij} &= 0 \quad \text{for all } K \in \mathcal{T}, \quad 1 \leq i, j \leq d, \\ \omega_{\mathcal{J},\sigma} \hat{\varepsilon}_{\sigma}^i - \hat{\lambda}_{\sigma}^i &= 0 \quad \text{for all } \sigma \in \mathcal{F}_{int}, \quad 1 \leq i \leq d, \\ \delta_{\mathcal{J}}^{\eta} \hat{\eta}_{K,\sigma}^i + \hat{\lambda}_K^{0,i} + \sum_{j=1}^d \hat{\lambda}_K^{1,ij} (x_{\sigma}^j - x_K^j) + \hat{\lambda}_{\sigma}^i &= 0 \quad \text{for all } K \in \mathcal{T}, \quad \sigma \in \mathcal{F}_K \cap \mathcal{F}_{int}, \quad 1 \leq i \leq d, \\ \delta_{\mathcal{J}}^{\eta} \hat{\eta}_{K,\sigma}^i + \hat{\lambda}_K^{0,i} + \sum_{j=1}^d \hat{\lambda}_K^{1,ij} (x_{\sigma}^j - x_K^j) &= 0 \quad \text{for all } K \in \mathcal{T}, \quad \sigma \in \mathcal{F}_K \cap \mathcal{F}_{ext}, \quad 1 \leq i \leq d, \\ \delta_{\mathcal{J}}^m \hat{m}_K + \hat{\lambda}_{\Omega} - \sum_{i=1}^d \hat{\lambda}_K^{1,ii} &= 0 \quad \text{for all } K \in \mathcal{T}, \\ \sum_{K \in \mathcal{T}} \hat{m}_K &= 0, \\ \sum_{\sigma \in \mathcal{F}_K} \hat{\eta}_{K,\sigma}^i &= \hat{\varepsilon}_{m,K}^{0,i} \quad \text{for all } K \in \mathcal{T}, \quad 1 \leq i \leq d, \\ \sum_{\sigma \in \mathcal{F}_K} \hat{\eta}_{K,\sigma}^i (x_{\sigma}^j - x_K^j) - \delta_{ij} \hat{m}_K &= \hat{\varepsilon}_{m,K}^{1,ij} \quad \text{for all } K \in \mathcal{T}, \quad 1 \leq i, j \leq d, \\ \sum_{K \in \mathcal{T}_{\sigma}} \hat{\eta}_{K,\sigma}^i &= \hat{\varepsilon}_{\sigma}^i \quad \text{for all } \sigma \in \mathcal{F}_{int}, \quad 1 \leq i \leq d. \end{aligned}$$

Let us denote $\hat{\lambda}_K^i(\mathbf{x}) = \hat{\lambda}_K^{0,i} + \sum_{j=1}^d \hat{\lambda}_K^{1,ij} (x^j - x_K^j)$. Notice that the first two sets of equations now implies $\hat{\lambda}_K^i(\mathbf{x}_{\sigma}) = -\hat{\lambda}_{\sigma}^i - \delta_{\mathcal{J}}^{\eta} \hat{\eta}_{K,\sigma}^i$ for all $\sigma \in \mathcal{F}_K$ (defining $\hat{\lambda}_{\sigma}^i = 0$ for any $\sigma \in \mathcal{F}_{ext}$). Further notice that the last set of equations on the $\hat{\eta}_{K,\sigma}$ implies that:

$$\begin{aligned} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d \hat{\eta}_{K,\sigma}^i \hat{\lambda}_K^i(\mathbf{x}_{\sigma}) &= \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d \hat{\eta}_{K,\sigma}^i (\hat{\lambda}_K^i(\mathbf{x}_{\sigma}) - \hat{\lambda}_K^{0,i}) + \sum_{i=1}^d \hat{\varepsilon}_{m,K}^{0,i} \hat{\lambda}_K^{0,i} \\ &= \sum_{i=1}^d \sum_{j=1}^d \hat{\lambda}_K^{1,ij} \left(\sum_{\sigma \in \mathcal{F}_K} \hat{\eta}_{K,\sigma}^i (x_{\sigma}^j - x_K^j) \right) + \sum_{i=1}^d \hat{\varepsilon}_{m,K}^{0,i} \hat{\lambda}_K^{0,i} = \hat{m}_K \operatorname{div}(\hat{\lambda}) + \sum_{i=1}^d \hat{\varepsilon}_{m,K}^{0,i} \hat{\lambda}_K^{0,i} + \sum_{i=1}^d \sum_{j=1}^d \hat{\varepsilon}_{m,K}^{1,ij} \hat{\lambda}_K^{1,ij} \\ &= \hat{m}_K \operatorname{div}(\hat{\lambda}) + \sum_{i=1}^d \omega_{\mathcal{J},K}^0 (\hat{\varepsilon}_{m,K}^{0,i})^2 + \sum_{i=1}^d \sum_{j=1}^d \omega_{\mathcal{J},K}^1 (\hat{\varepsilon}_{m,K}^{1,ij})^2. \end{aligned}$$

Thus, summing over $K \in \mathcal{T}$, we get:

$$\sum_{K \in \mathcal{T}} \hat{m}_K \operatorname{div}(\hat{\lambda}_K) = \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d \hat{\eta}_{K,\sigma}^i \hat{\lambda}_K^i(\mathbf{x}_{\sigma}) - \sum_{i=1}^d \omega_{\mathcal{J},K}^0 (\hat{\varepsilon}_{m,K}^{0,i})^2 - \sum_{i=1}^d \sum_{j=1}^d \omega_{\mathcal{J},K}^1 (\hat{\varepsilon}_{m,K}^{1,ij})^2$$

$$\begin{aligned}
&= - \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d (\hat{\eta}_{K,\sigma}^i \hat{\lambda}_\sigma^i + \delta_{\mathcal{J}}^\eta (\hat{\eta}_{K,\sigma}^i)^2) - \sum_{i=1}^d \omega_{\mathcal{J},K}^0 (\hat{\varepsilon}_{m,K}^{0,i})^2 - \sum_{i=1}^d \sum_{j=1}^d \omega_{\mathcal{J},K}^1 (\hat{\varepsilon}_{m,K}^{1,ij})^2 \\
&= -\delta_{\mathcal{J}}^\eta \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d (\hat{\eta}_{K,\sigma}^i)^2 - \sum_{\sigma \in \mathcal{F}_{int}} \sum_{i=1}^d \omega_{\mathcal{J},\sigma} (\hat{\varepsilon}_\sigma^i)^2 - \sum_{i=1}^d \omega_{\mathcal{J},K}^0 (\hat{\varepsilon}_{m,K}^{0,i})^2 - \sum_{i=1}^d \sum_{j=1}^d \omega_{\mathcal{J},K}^1 (\hat{\varepsilon}_{m,K}^{1,ij})^2 \leq 0,
\end{aligned}$$

as $\sum_{K \in \mathcal{T}_\sigma} \hat{\eta}_{K,\sigma}^i = \hat{\varepsilon}_\sigma^i$ for $\sigma \in \mathcal{F}_{int}$ and $\hat{\lambda}_\sigma^i = 0$ for $\sigma \in \mathcal{F}_{ext}$. Multiplying

$$\delta_{\mathcal{J}}^m \hat{m}_K + \hat{\lambda}_\Omega - \sum_{i=1}^d \hat{\lambda}_K^{1,ii} = 0 \quad \text{for all } K \in \mathcal{T}$$

by \hat{m}_K and summing over $K \in \mathcal{T}$, we thus get as $\text{div}(\hat{\lambda}_K) = \sum_{i=1}^d \hat{\lambda}_K^{1,ii}$ and $\sum_{K \in \mathcal{T}} \hat{m}_K = 0$:

$$\begin{aligned}
&\sum_{K \in \mathcal{T}} \left(\delta_{\mathcal{J}}^m \hat{m}_K^2 + \hat{m}_K \hat{\lambda}_\Omega - \hat{m}_K \sum_{i=1}^d \hat{\lambda}_K^{1,ii} \right) = \delta_{\mathcal{J}}^m \sum_{K \in \mathcal{T}} \hat{m}_K^2 - \sum_{K \in \mathcal{T}} \hat{m}_K \sum_{i=1}^d \hat{\lambda}_K^{1,ii} = \delta_{\mathcal{J}}^m \sum_{K \in \mathcal{T}} \hat{m}_K^2 - \sum_{K \in \mathcal{T}} \hat{m}_K \text{div}(\hat{\lambda}_K) \\
&= \delta_{\mathcal{J}}^m \sum_{K \in \mathcal{T}} \hat{m}_K^2 + \delta_{\mathcal{J}}^\eta \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d (\hat{\eta}_{K,\sigma}^i)^2 + \sum_{\sigma \in \mathcal{F}_{int}} \sum_{i=1}^d \omega_{\mathcal{J},\sigma} (\hat{\varepsilon}_\sigma^i)^2 + \sum_{i=1}^d \omega_{\mathcal{J},K}^0 (\hat{\varepsilon}_{m,K}^{0,i})^2 + \sum_{i=1}^d \sum_{j=1}^d \omega_{\mathcal{J},K}^1 (\hat{\varepsilon}_{m,K}^{1,ij})^2 = 0,
\end{aligned}$$

immediately implying that $\hat{m}_K = 0$ for all $K \in \mathcal{T}$, $\hat{\eta}_{K,\sigma}^i = 0$ for all $K \in \mathcal{T}$, $\sigma \in \mathcal{F}_K$ and $1 \leq i \leq d$ and $\hat{\varepsilon}_{m,K}^{0,i} = 0$ and $\hat{\varepsilon}_{m,K}^{1,ij} = 0$ for all $K \in \mathcal{T}$ and $1 \leq i, j \leq d$ as $\delta_{\mathcal{J}}^m > 0$ and $\delta_{\mathcal{J}}^\eta > 0$ and $\hat{\varepsilon}_\sigma^i = 0$ for all $\sigma \in \mathcal{F}_{int}$. Consequently, using the equations of the system we also get $\hat{\lambda}_\Omega = 0$, $\hat{\lambda}_\sigma^i = 0$ for all $\sigma \in \mathcal{F}_{int}$ and all $1 \leq i \leq d$ as well as $\hat{\lambda}_K^{0,i} = 0$ and $\hat{\lambda}_K^{1,ij} = 0$ for all $K \in \mathcal{T}$ and $1 \leq i, j \leq d$. Thus $\text{Ker } \mathbb{A}_{\mathcal{J}}^T = \{0\}$ and the system is always invertible.

Appendix B. Proof of lemma 4.2

Consider the sub-matrix of $\mathbb{A}_{\mathcal{J}}$ that correspond to the equations on $(\eta, \varepsilon(\mathcal{G}), \lambda)$ (thus excluding the equations on $(m_K)_{K \in \mathcal{T}}$ and λ_Ω). This corresponds to the equations:

$$\begin{aligned}
&\omega_{\mathcal{J},\sigma} \varepsilon_\sigma^i - \lambda_\sigma^i = 0 \quad \text{for all } \sigma \in \mathcal{F}_{int}, \quad 1 \leq i \leq d, \\
&\omega_{\mathcal{J},K}^0 \varepsilon_{m,K}^{0,i} - \lambda_K^{0,i} = 0 \quad \text{for all } K \in \mathcal{T}, \quad 1 \leq i \leq d, \\
&\omega_{\mathcal{J},K}^1 \varepsilon_{m,K}^{1,ij} - \lambda_K^{1,ij} = 0 \quad \text{for all } K \in \mathcal{T}, \quad 1 \leq i, j \leq d, \\
&\delta_{\mathcal{J}}^\eta \eta_{K,\sigma}^i + \sum_{j=1}^d \lambda_K^{1,ij} (x_\sigma^j - x_K^j) + \lambda_K^{0,i} + \lambda_\sigma^i = 0 \quad \text{for all } K \in \mathcal{T}, \quad \sigma \in \mathcal{F}_K, \quad 1 \leq i \leq d, \\
&\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i = \varepsilon_{m,K}^{0,i} \quad \text{for all } K \in \mathcal{T} \quad \text{and all } 1 \leq i \leq d, \\
&\sum_{\sigma \in \mathcal{F}_K} \eta_{K,\sigma}^i (x_\sigma^j - x_K^j) = m_K \delta_{ij} + \varepsilon_{m,K}^{1,ij} \quad \text{for all } K \in \mathcal{T} \quad \text{and all } 1 \leq i, j \leq d, \\
&\sum_{K \in \mathcal{T}_\sigma} \eta_{K,\sigma}^i = \varepsilon_\sigma^i \quad \text{for all } \sigma \in \mathcal{F}_{int} \quad \text{and all } 1 \leq i \leq d.
\end{aligned}$$

Using the symmetry of the system or proceeding as before, it is straightforward to see that elements in the null space of the adjoint of this sub-system will satisfy:

$$\begin{aligned}
&\omega_{\mathcal{J},\sigma} \hat{\varepsilon}_\sigma^i - \hat{\lambda}_\sigma^i = 0 \quad \text{for all } \sigma \in \mathcal{F}_{int}, \quad 1 \leq i \leq d, \\
&\omega_{\mathcal{J},K}^0 \hat{\varepsilon}_{m,K}^{0,i} - \hat{\lambda}_K^{0,i} = 0 \quad \text{for all } K \in \mathcal{T}, \quad 1 \leq i \leq d,
\end{aligned}$$

$$\begin{aligned}
\omega_{\mathcal{J},K}^1 \hat{\varepsilon}_{m,K}^{1,ij} - \hat{\lambda}_K^{1,ij} &= 0 \quad \text{for all } K \in \mathcal{T}, \quad 1 \leq i, j \leq d, \\
\delta_{\mathcal{J}}^{\eta} \hat{\eta}_{K,\sigma}^i + \sum_{j=1}^d \hat{\lambda}_K^{1,ij} (x_{\sigma}^j - x_K^j) + \hat{\lambda}_K^{0,i} + \hat{\lambda}_{\sigma}^i &= 0 \quad \text{for all } K \in \mathcal{T}, \quad \sigma \in \mathcal{F}_K, \quad 1 \leq i \leq d, \\
\sum_{\sigma \in \mathcal{F}_K} \hat{\eta}_{K,\sigma}^i &= \hat{\varepsilon}_{m,K}^{0,i} \quad \text{for all } K \in \mathcal{T} \quad \text{and all } 1 \leq i \leq d, \\
\sum_{\sigma \in \mathcal{F}_K} \hat{\eta}_{K,\sigma}^i (x_{\sigma}^j - x_K^j) &= \hat{\varepsilon}_{m,K}^{1,ij} \quad \text{for all } K \in \mathcal{T} \quad \text{and all } 1 \leq i, j \leq d, \\
\sum_{K \in \mathcal{T}_{\sigma}} \hat{\eta}_{K,\sigma}^i &= \hat{\varepsilon}_{\sigma}^i \quad \text{for all } \sigma \in \mathcal{F}_{int} \quad \text{and all } 1 \leq i \leq d.
\end{aligned}$$

We have:

$$\sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d \hat{\eta}_{K,\sigma}^i \hat{\lambda}_K^i(\mathbf{x}_{\sigma}) = - \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d (\hat{\eta}_{K,\sigma}^i \hat{\lambda}_{\sigma}^i + \delta_{\mathcal{J}}^{\eta} (\hat{\eta}_{K,\sigma}^i)^2) = - \sum_{\sigma \in \mathcal{F}_{int}} \sum_{i=1}^d \omega_{\mathcal{J},\sigma} (\hat{\varepsilon}_{\sigma}^i)^2 - \delta_{\mathcal{J}}^{\eta} \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d (\hat{\eta}_{K,\sigma}^i)^2 \leq 0,$$

as $\sum_{K \in \mathcal{T}_{\sigma}} \hat{\eta}_{K,\sigma}^i = \hat{\varepsilon}_{\sigma}^i$ for $\sigma \in \mathcal{F}_{int}$ and $\hat{\varepsilon}_{\sigma}^i = 0$ for $\sigma \in \mathcal{F}_{ext}$. Meanwhile, we see that:

$$\begin{aligned}
\sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d \hat{\eta}_{K,\sigma}^i \hat{\lambda}_K^i(\mathbf{x}_{\sigma}) &= \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d \hat{\eta}_{K,\sigma}^i (\hat{\lambda}_K^i(\mathbf{x}_{\sigma}) - \hat{\lambda}_K^{0,i}) + \sum_{i=1}^d \hat{\varepsilon}_{m,K}^{0,i} \hat{\lambda}_K^{0,i} \\
&= \sum_{i=1}^d \sum_{j=1}^d \hat{\lambda}_K^{1,ij} \left(\sum_{\sigma \in \mathcal{F}_K} \hat{\eta}_{K,\sigma}^i (x_{\sigma}^j - x_K^j) \right) + \sum_{i=1}^d \hat{\varepsilon}_{m,K}^{0,i} \hat{\lambda}_K^{0,i} = \sum_{i=1}^d \hat{\varepsilon}_{m,K}^{0,i} \hat{\lambda}_K^{0,i} + \sum_{i=1}^d \sum_{j=1}^d \hat{\varepsilon}_{m,K}^{1,ij} \hat{\lambda}_K^{1,ij} \\
&= \sum_{i=1}^d \omega_{\mathcal{J},K}^0 (\hat{\varepsilon}_{m,K}^{0,i})^2 + \sum_{i=1}^d \sum_{j=1}^d \omega_{\mathcal{J},K}^1 (\hat{\varepsilon}_{m,K}^{1,ij})^2,
\end{aligned}$$

and thus summing over $K \in \mathcal{T}$, we get:

$$\sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d \hat{\eta}_{K,\sigma}^i \hat{\lambda}_K^i(\mathbf{x}_{\sigma}) = \sum_{K \in \mathcal{T}} \sum_{i=1}^d \omega_{\mathcal{J},K}^0 (\hat{\varepsilon}_{m,K}^{0,i})^2 + \sum_{K \in \mathcal{T}} \sum_{i=1}^d \sum_{j=1}^d \omega_{\mathcal{J},K}^1 (\hat{\varepsilon}_{m,K}^{1,ij})^2 \geq 0.$$

We immediately deduce that $\hat{\eta}_{K,\sigma}^i = 0$ for all $K \in \mathcal{T}$, all $\sigma \in \mathcal{F}_K$ and all $1 \leq i \leq d$, that $\hat{\varepsilon}_{m,K}^{0,i} = 0$ and $\hat{\varepsilon}_{m,K}^{1,ij} = 0$ for all $K \in \mathcal{T}$ and all $1 \leq i, j \leq d$, which in turn implies that $\hat{\lambda}_K^{0,i} = 0$ and $\hat{\lambda}_K^{1,ij} = 0$ for all $K \in \mathcal{T}$ and all $1 \leq i, j \leq d$, as well as $\hat{\lambda}_{\sigma}^i = 0$ and thus also $\hat{\varepsilon}_{\sigma}^i = 0$ for all $\sigma \in \mathcal{F}_{int}$, $1 \leq i \leq d$. The sub-system is consequently always invertible, therefore its solution can be written under the form (since λ_{Ω} does not appear in the sub-system's second member):

$$(\boldsymbol{\eta}, \boldsymbol{\varepsilon}(\mathcal{G}), \lambda)^T = \mathbb{B} \mathbf{m},$$

for some matrix \mathbb{B} independent on $\delta_{\mathcal{J}}^m$. In particular, let us denote $\lambda_K^{1,ii} = \mathbb{B}_K^{1,ii} \mathbf{m}$. The solution of (15) thus satisfies:

$$\sum_{K \in \mathcal{T}} m_K = - \frac{\text{card}(\mathcal{T})}{\delta_{\mathcal{J}}^m} \lambda_{\Omega} + \frac{1}{\delta_{\mathcal{J}}^m} \sum_{K \in \mathcal{T}} \sum_{i=1}^d \lambda_K^{1,ii} = - \frac{\text{card}(\mathcal{T})}{\delta_{\mathcal{J}}^m} \lambda_{\Omega} + \frac{1}{\delta_{\mathcal{J}}^m} \sum_{K \in \mathcal{T}} \sum_{i=1}^d \mathbb{B}_K^{1,ii} \mathbf{m} = |\Omega|,$$

and thus:

$$\lambda_{\Omega} = \frac{1}{\text{card}(\mathcal{T})} \sum_{K \in \mathcal{T}} \sum_{i=1}^d \mathbb{B}_K^{1,ii} \mathbf{m} - \frac{\delta_{\mathcal{J}}^m}{\text{card}(\mathcal{T})} |\Omega|.$$

Injecting this into the equation on m_K , we get:

$$\delta_{\mathcal{J}}^m m_K + \frac{1}{\text{card}(\mathcal{T})} \sum_{L \in \mathcal{T}} \sum_{i=1}^d \mathbb{B}_L^{1,ii} \mathbf{m} - \sum_{i=1}^d \mathbb{B}_K^{1,ii} \mathbf{m} - \frac{\delta_{\mathcal{J}}^m}{\text{card}(\mathcal{T})} |\Omega| = 0.$$

This rewrites:

$$\left(\mathbb{I}_d - \frac{1}{\delta_{\mathcal{J}}^m} \tilde{\mathbb{B}} \right) \mathbf{m} = \frac{|\Omega|}{\text{card}(\mathcal{T})} \mathbf{1},$$

with $\mathbf{1}$ the vector with every component equal to one. If $\delta_0 > 0$ denotes the spectral radius of $\tilde{\mathbb{B}}$ (which is finite as the dimension is finite) then for $\delta_{\mathcal{J}}^m > \delta_0$ the spectral radius of $\frac{1}{\delta_{\mathcal{J}}^m} \tilde{\mathbb{B}}$ is strictly below one. Thus the inverse matrix of $\mathbb{I}_d - \frac{1}{\delta_{\mathcal{J}}^m} \tilde{\mathbb{B}}$ is given by the Neumann series:

$$\mathbf{m} = \left(\mathbb{I}_d + \sum_{k=1}^{+\infty} \frac{1}{(\delta_{\mathcal{J}}^m)^k} \tilde{\mathbb{B}}^k \right) \frac{|\Omega|}{\text{card}(\mathcal{T})} \mathbf{1}.$$

In particular, this implies that $\lim_{\delta_{\mathcal{J}}^m \rightarrow +\infty} m_K = \frac{|\Omega|}{\text{card}(\mathcal{T})}$, from which we immediately deduce the result.

Appendix C. Proof of lemma 4.3

Symmetry can be easily established proceeding as in the proof of lemma 4.1. Consequently, for $\mathbb{A}_{\mathcal{J},r}^T \hat{\mathbf{g}}_r = 0$ we deduce that:

$$\delta_{\mathcal{J}}^n \hat{\eta}_{K,\sigma}^i + \hat{\lambda}_K^{0,i} + \sum_{j=1}^d \hat{\lambda}_K^{1,ij} (x_{\sigma}^j - x_K^j) + \hat{\lambda}_{\sigma}^i = 0 \quad \text{for all } K \in \mathcal{T}, \sigma \in \mathcal{F}_K \cap \mathcal{F}_{int}, 1 \leq i \leq d,$$

$$\delta_{\mathcal{J}}^n \hat{\eta}_{K,\sigma}^i + \hat{\lambda}_K^{0,i} + \sum_{j=1}^d \hat{\lambda}_K^{1,ij} (x_{\sigma}^j - x_K^j) = 0 \quad \text{for all } K \in \mathcal{T}, \sigma \in \mathcal{F}_K \cap \mathcal{F}_{ext}, 1 \leq i \leq d,$$

$$\delta_{\mathcal{J}}^m \hat{m}_K + \hat{\lambda}_{\Omega} - \sum_{i=1}^d \hat{\lambda}_K^{1,ii} = 0 \quad \text{for all } K \in \mathcal{T},$$

$$\sum_{K \in \mathcal{T}} \hat{m}_K = 0,$$

$$\sum_{\sigma \in \mathcal{F}_K} \hat{\eta}_{K,\sigma}^i = 0 \quad \text{for all } K \in \mathcal{T}, 1 \leq i \leq d,$$

$$\sum_{\sigma \in \mathcal{F}_K} \hat{\eta}_{K,\sigma}^i (x_{\sigma}^j - x_K^j) - \delta_{ij} \hat{m}_K = 0 \quad \text{for all } K \in \mathcal{T}, 1 \leq i, j \leq d,$$

$$\sum_{K \in \mathcal{T}_{\sigma}} \hat{\eta}_{K,\sigma}^i = 0 \quad \text{for all } \sigma \in \mathcal{F}_{int}, 1 \leq i \leq d.$$

Let us denote $\hat{\lambda}_K^i(\mathbf{x}) = \hat{\lambda}_K^{0,i} + \sum_{j=1}^d \hat{\lambda}_K^{1,ij} (x^j - x_K^j)$. Notice that the first two sets of equations now implies $\hat{\lambda}_K^i(\mathbf{x}_{\sigma}) = -\hat{\lambda}_{\sigma}^i - \delta_{\mathcal{J}}^n \hat{\eta}_{K,\sigma}^i$ for all $\sigma \in \mathcal{F}_K$ (defining $\hat{\lambda}_{\sigma}^i = 0$ for any $\sigma \in \mathcal{F}_{ext}$). Further notice that the last set of equations on the $\hat{\eta}_{K,\sigma}$'s implies that:

$$\sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d \hat{\eta}_{K,\sigma}^i \hat{\lambda}_K^i(\mathbf{x}_{\sigma}) = \hat{m}_K \left(\sum_{i=1}^d \hat{\lambda}_K^{1,ii} \right) = \hat{m}_K \text{div}(\hat{\lambda}_K).$$

Thus, summing over $K \in \mathcal{T}$, we get:

$$\begin{aligned} \sum_{K \in \mathcal{T}} \hat{m}_K \operatorname{div}(\hat{\lambda}_K) &= \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d \hat{\eta}_{K,\sigma}^i \hat{\lambda}_K^i(\mathbf{x}_\sigma) \\ &= - \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d (\hat{\eta}_{K,\sigma}^i \hat{\lambda}_\sigma^i + \delta_{\mathcal{J}}^\eta (\hat{\eta}_{K,\sigma}^i)^2) = -\delta_{\mathcal{J}}^\eta \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d (\hat{\eta}_{K,\sigma}^i)^2, \end{aligned}$$

as $\sum_{K \in \mathcal{T}_\sigma} \hat{\eta}_{K,\sigma}^i = 0$ for $\sigma \in \mathcal{F}_{int}$ and $\hat{\lambda}_\sigma^i = 0$ for $\sigma \in \mathcal{F}_{ext}$. Multiplying

$$\delta_{\mathcal{J}}^m \hat{m}_K + \hat{\lambda}_\Omega - \sum_{i=1}^d \hat{\lambda}_K^{1,ii} = 0 \quad \text{for all } K \in \mathcal{T},$$

by \hat{m}_K and summing over $K \in \mathcal{T}$, we thus get

$$\begin{aligned} \sum_{K \in \mathcal{T}} \left(\delta_{\mathcal{J}}^m \hat{m}_K^2 + \hat{m}_K \hat{\lambda}_\Omega - \hat{m}_K \sum_{i=1}^d \hat{\lambda}_K^{1,ii} \right) &= \delta_{\mathcal{J}}^m \sum_{K \in \mathcal{T}} \hat{m}_K^2 - \sum_{K \in \mathcal{T}} \hat{m}_K \sum_{i=1}^d \hat{\lambda}_K^{1,ii} = \delta_{\mathcal{J}}^m \sum_{K \in \mathcal{T}} \hat{m}_K^2 - \sum_{K \in \mathcal{T}} \hat{m}_K \operatorname{div}(\hat{\lambda}_K) \\ &= \delta_{\mathcal{J}}^m \sum_{K \in \mathcal{T}} \hat{m}_K^2 + \delta_{\mathcal{J}}^\eta \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d (\hat{\eta}_{K,\sigma}^i)^2 = 0, \end{aligned}$$

immediately implying that $\hat{m}_K = 0$ for all $K \in \mathcal{T}$ and $\hat{\eta}_{K,\sigma}^i = 0$ for all $K \in \mathcal{T}$, $\sigma \in \mathcal{F}_K$ and $1 \leq i \leq d$. Consequently, we also get $\hat{\lambda}_\Omega = 0$. As

$$\mathbf{b}_r^T \hat{\mathbf{g}}_r = |\Omega| \hat{\lambda}_\Omega,$$

we get $\mathbf{b}_r^T \hat{\mathbf{g}}_r = 0$ for all $\hat{\mathbf{g}}_r$ such that $\mathbb{A}_{\mathcal{J},r}^T \hat{\mathbf{g}}_r = 0$ and the system has a solution from Fredholm's alternative.

Now, for any $\hat{\mathbf{g}}_r$ such that $\mathbb{A}_{\mathcal{J},r}^T \hat{\mathbf{g}}_r = 0$, there remains the conditions:

$$\hat{\lambda}_K^{0,i} + \sum_{j=1}^d \hat{\lambda}_K^{1,ij} (x_\sigma^j - x_K^j) + \hat{\lambda}_\sigma^i = 0 \quad \text{for all } K \in \mathcal{T}, \sigma \in \mathcal{F}_K \cap \mathcal{F}_{int}, 1 \leq i \leq d,$$

$$\hat{\lambda}_K^{0,i} + \sum_{j=1}^d \hat{\lambda}_K^{1,ij} (x_\sigma^j - x_K^j) = 0 \quad \text{for all } K \in \mathcal{T}, \sigma \in \mathcal{F}_K \cap \mathcal{F}_{ext}, 1 \leq i \leq d.$$

From hypothesis (H), we deduce that this implies $\hat{\lambda}_K^{0,i} = 0$ for all $K \in \mathcal{T}$ and all $1 \leq i \leq d$ and $\hat{\lambda}_K^{1,ij} = 0$ for all $K \in \mathcal{T}$ and all $1 \leq i, j \leq d$. Thus, $\operatorname{Ker} \mathbb{A}_{\mathcal{J},r}^T = \operatorname{Ker} \mathbb{A}_{\mathcal{J},r} = \{0\}$ and the system is invertible.

Appendix D. Proof of lemma 4.4

Proceeding as in the non-exact case, consider the sub-matrix of $\mathbb{A}_{\mathcal{J},r}$ that correspond to the equations on $(\boldsymbol{\eta}, \lambda)$ (thus excluding the equations on $(m_K)_{K \in \mathcal{T}}$ and λ_Ω). Again, using the symmetry of the system or proceeding as before, it is straightforward to see that elements in the null space of the adjoint of this sub-system will satisfy:

$$\delta_{\mathcal{J}}^\eta \hat{\eta}_{K,\sigma}^i + \sum_{j=1}^d \hat{\lambda}_K^{1,ij} (x_\sigma^j - x_K^j) + \hat{\lambda}_K^{0,i} + \hat{\lambda}_\sigma^i = 0 \quad \text{for all } K \in \mathcal{T}, \sigma \in \mathcal{F}_K, 1 \leq i \leq d,$$

$$\sum_{\sigma \in \mathcal{F}_K} \hat{\eta}_{K,\sigma}^i = 0 \quad \text{for all } K \in \mathcal{T} \text{ and all } 1 \leq i \leq d,$$

$$\sum_{\sigma \in \mathcal{F}_K} \hat{\eta}_{K,\sigma}^i (x_\sigma^j - x_K^j) = 0 \quad \text{for all } K \in \mathcal{T} \text{ and all } 1 \leq i, j \leq d,$$

$$\sum_{K \in \mathcal{T}_\sigma} \hat{\eta}_{K\sigma}^i = 0 \quad \text{for all } \sigma \in \mathcal{F}_{int} \text{ and all } 1 \leq i \leq d.$$

We have:

$$\sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d \hat{\eta}_{K,\sigma}^i \hat{\lambda}_K^i(\mathbf{x}_\sigma) = - \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d (\hat{\eta}_{K,\sigma}^i \hat{\epsilon}_\sigma^i + \delta_{\mathcal{J}}^\eta (\hat{\eta}_{K,\sigma}^i)^2) = -\delta_{\mathcal{J}}^\eta \sum_{K \in \mathcal{T}} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d (\hat{\eta}_{K,\sigma}^i)^2 \leq 0,$$

as $\sum_{K \in \mathcal{T}_\sigma} \hat{\eta}_{K,\sigma}^i = 0$ for $\sigma \in \mathcal{F}_{int}$ and $\hat{\lambda}_\sigma^i = 0$ for $\sigma \in \mathcal{F}_{ext}$. Meanwhile, we see that:

$$\begin{aligned} \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d \hat{\eta}_{K,\sigma}^i \hat{\lambda}_K^i(\mathbf{x}_\sigma) &= \sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d \hat{\eta}_{K,\sigma}^i (\hat{\lambda}_K^i(\mathbf{x}_\sigma) - \hat{\lambda}_K^{0,i}), \\ &= \sum_{i=1}^d \sum_{j=1}^d \hat{\lambda}_K^{1,ij} \left(\sum_{\sigma \in \mathcal{F}_K} \hat{\eta}_{K,\sigma}^i (x_\sigma^j - x_K^j) \right) = 0, \end{aligned}$$

and thus summing over $K \in \mathcal{T}$, we get:

$$\sum_{\sigma \in \mathcal{F}_K} \sum_{i=1}^d \hat{\eta}_{K,\sigma}^i \hat{\lambda}_K^i(\mathbf{x}_\sigma) = 0.$$

We immediately deduce that $\hat{\eta}_{K,\sigma}^i = 0$ for all $K \in \mathcal{T}$, all $\sigma \in \mathcal{F}_K$ and all $1 \leq i \leq d$, which in turn implies that

$$\sum_{j=1}^d \hat{\lambda}_K^{1,ij} (x_\sigma^j - x_K^j) + \hat{\lambda}_K^{0,i} + \hat{\lambda}_\sigma^i = 0 \quad \text{for all } K \in \mathcal{T}, \sigma \in \mathcal{F}_K \cap \mathcal{F}_{int}, 1 \leq i \leq d,$$

and

$$\sum_{j=1}^d \hat{\lambda}_K^{1,ij} (x_\sigma^j - x_K^j) + \hat{\lambda}_K^{0,i} = 0 \quad \text{for all } K \in \mathcal{T}, \sigma \in \mathcal{F}_K \cap \mathcal{F}_{ext}, 1 \leq i \leq d.$$

Under hypothesis (H), we know that this implies that $\hat{\lambda}_K^{0,i} = 0$ and $\hat{\lambda}_K^{1,ij} = 0$ for all $K \in \mathcal{T}$ and all $1 \leq i, j \leq d$, as well as $\hat{\lambda}_\sigma^i = 0$ for all $\sigma \in \mathcal{F}_{int}$, $1 \leq i \leq d$. The sub-system is consequently always invertible, therefore its solution can be written under the form (since λ_Ω does not appear in the sub-system's second member):

$$(\boldsymbol{\eta}, \boldsymbol{\lambda})^T = \mathbb{B}\mathbf{m},$$

and the remaining of the proof is now identical to the end of the proof of lemma 4.2.