Proceedings of the Eighth International Conference on
Parallel, Distributed, GPU and Cloud Computing for Engineering
Edited by: P. Iványi, J. Kruis and B.H.V. Topping
Civil-Comp Conferences, Volume 12, Paper 2.3
Civil-Comp Press, Edinburgh, United Kingdom, 2025
ISSN: 2753-3239, doi: 10.4203/ccc.12.2.3

An Enhanced Spectral Method for Mesh Partitioning Based on Matching Edges

T. Fouque[1], C.-H. Lai[2], E. George[2] and F. Magoulès[1]

[1] MICS, CentraleSupélec, Université Paris-Saclay, Gif-sur-Yvette, Île-de-France, France
[2] School of Computing and Mathematical Sciences, University of Greenwich, London, United Kingdom

Abstract

In this article, we present an original method for mesh partitioning based on a multi-level approach followed by a spectral method applied to the coarsened graph. Opposite to other mesh partitioning algorithms based on spectral method, our approach takes into account the number of nodes as a parameter in the coarsened graph, ensuring a higher quality of the partitions. In the case of planar meshes, we prove some theoretical results of this new coarse spectral bisection method related to this partitioning. The numerical experiments confirm the efficiency of this method on different test cases.

Keywords: finite element method, mesh partitioning, spectral method, planar graph, parallel computing, domain decomposition.

1 Introduction

When considering large scale problems, parallel algorithms are mandatory both to reduce the computational time of the simulation and also to ensure distributed data

through the processors. Domain Decomposition methods are well suited for parallel computations. Indeed, the division of a problem into smaller subproblems, through artificial subdivisions of the domain, is a mean for introducing parallelism [13] [12]. The implementation of such methods on a parallel computer is strongly dependent of the communications between the processors, which depend on the interface size between the subdomains, which impacts the size of the array exchanged between the processors. The load-balancing between the subdomains impacts the computational time spent to solve the local subproblems in each subdomain. When bad load balancing appears, the latter point implies delay in the synchronization between the processors. Opposite when using asynchronous domain decomposition methods, like the one considered in [8] [9] [5], bad load balancing is not an issue anymore. However, the time to generate a good quality partitioning still remains important. Within the partitioning algorithms, several approaches exist like [11], spectral methods [2], or hybrid methods [16] [6]. It is a well known that this kind of problem is NP-complete [1]. Thus, this problem is mainly solved using heuristics such as the algorithms implemented in JOSTLE [17], METIS, or HMETIS [11]. Coarsening methods have been used for decades and were initially introduced because of the large dimension of the graph. Most of the software like JOSTLE or METIS are based on coarsening approaches.

In this paper, we study an adapted spectral method for coarse graphs. The difference with [2] is that the method presented by the authors does not take into account the way the graph is coarsened and mainly relies on using heuristics during the uncoarsening procedure. Thus, we will encode the information at the coarsening level and modify the spectral problem to solve. We do not have to apply heuristics in the uncoarsening procedure. We will show that the coarse graph still respects some desirable theoretical properties, and illustrates these properties in the numerical experiments.

2 Formalisation of the problem

We are given a mesh $\mathcal{M}$, issued for example, from a Delaunay triangulation process, a common method to generate triangular meshes [3] [14].

Throughout this paper, we work with the dual graph of the finite element $G = (V, E)$ where $E \subset V^2$. It is an unweighted and non-directed graph. In the case of 2D meshes, the graph has an additional propriety : it is planar, i.e. the graph can be plot in the 2D plane in such a way that no edges crosses. We will denote by n the number of nodes in the graph and we will set $\Delta \in \mathbb{N}^*$ being the maximal degree in the graph, each node has a degree at most Δ. For $P \subset V$, we denote by $|P|$ its cardinality and by $\delta(P)$ the set of edges having exactly one extremity in P. More formally it can be defined as : $\delta(P) = \{(v, v') \in E | v \in P, v' \notin P\}$. We extend the definition to $\delta(P, P') = \{(v, v') \in E | v \in P, v' \in P'\}$, so that $\delta(P) = \delta(P, V \setminus P)$.

There are metrics we would to minimize when we are given a graph to partition. Let's denote $g \in \mathbb{N}^*$ the number of subdomains we aim to obtain and a partition $P_1, P_2, ..., P_g$ of V. The first two metrics defined below were first introduced in [7].

Definition 1 (Interface Size)**.** *We define the Interface Size (IS) as the number of edges cut between partitions :*

$$IS(P_1, P_2, ..., P_g) = \sum_{k=1}^{g} |\delta(P_k)|$$

Definition 2 (Load balancing)**.** *We define the Load Interface Balancing (LIB) which represents the balance of the number of nodes in each partition as :*

$$LIB(P_1, ..., P_g) = \left(\sum_{k=1}^{g} \left(|P_k| - \frac{n}{g} \right)^2 \right)^{\frac{1}{2}}.$$

The quantity LIB tends to 0 when the partitions become perfectly balanced.

Definition 3 (Isoperimetric Ratio)**.** *We define the isoperimetric ratio ($\phi_G(P)$) as the ratio between the interface size and the number of nodes inside a partition as :*

$$\phi_G(P) = \frac{|\delta(P)|}{min(|P|, |V \setminus P|)}$$

This will also be referred to as cut ratio. We define the Cheeger constant as : $h(G) = min_{S \subset V, |S| \leq n/2} \phi_G(S)$ which is a common measure of the connectivity of a graph. If this constant is high it means that the graph is difficult to partition.

3 Graph coarsening with edge matching

As mentioned in the introduction the problem of graph partitioning is NP-complete. To overcome difficulties related to exponential running time, unless P=NP, we reduce the size of the graph. We use a coarsening approach as it is done in METIS [11] or JOSTLE [17], relying on several passes of matching edges : at each step, we contract vertices to decrease the size of the graph. Algorithm 1 illustrates the complete process.

Algorithm 1 $MatchingEdge(G, n_{coarse})$

Ensure: G is a graph, and n_{coarse} is the desired coarsening level
1: $V' \leftarrow V$, $n_{iter} \leftarrow 0$
2: **while** $n_{iter} \neq n_{coarse}$ **do**
3: Find a matching of edges: $(v_1, v_2), \ldots, (v_k, v_{k+1})$
4: $V' \leftarrow$ Nodes formed by contraction of edges previously obtained
5: $n_{iter} \leftarrow n_{iter} + 1$
6: **end while**

We set $G_0 = G$. At each iteration the algorithm constructs a new coarsened graph G_k whose vertices are made of at most 2 nodes from G_{k-1}. To create G_k we do a

matching of the edges of G_{k-1}, in other words, we choose a set of edges: $E_k = \{(v_1, v_2), (v_3, v_4), \ldots, (v_m, v_{m+1})\}$ in G_{k-1} such that each vertex v of G_{k-1} belongs to at most one edge in E_k. We construct this matching based on a random procedure called Random Matching in METIS [11] or JOSTLE [17] which is the naive way to construct a matching in a graph. At the end of the algorithm, we get a final coarsened graph $G' := G_{ncoarse}$, whose vertices will be called metanode and can be seen as a subset of vertices from the original graph G_0. This final graph is also called the metanode graph.

Once this step of coarsening is done we apply a spectral method to partition the coarsened graph before decoarsening it to get the final partitioning. We will apply the same procedure as done in [10] to perform the spectral clustering on the metanode graph. Here we derive some properties of our metanode graph. A first property concerns the connectivity of our metanodes.

Proposition 1. *The metanode graph G' is connected.*

Proof. This proposition can be proven trivially by induction on the number of coarsification steps we do. At each iteration, two neighboring nodes v_1 and v_2 are contracted. As a result, the union of the sets of original nodes represented by v_1 and v_2 forms a connected component in the original graph G. □

In the rest of the paper we assume that the graph is planar, it means that we can represent it on the plane and no edges will cross. Most of 2D meshes are planar this justifies our assumption. Such planar graphs have very good properties we want to keep on the final coarsened graph to prove theoretical results about spectral method. We want to show that the metanode graph remains planar.

Proposition 2. *The metanode graph G' is planar.*

Proof. Thanks to proposition 1, if we show that contracting an edge in a planar graph keeps it planar we will have the result. This is a classic result, and the proof can be found in [4]. In fact, we need this result to show Kuratowski's theorem which states that a graph is planar if and only if neither K_5 nor $K_{3,3}$ are minors of it. □

4 Spectral method for mesh partitioning

4.1 Decomposition using spectral method

Now we review our optimization step using the spectral method. If we restrict ourselves to the case of two subdomains, one associated to -1 and the other to $+1$, we can search for a vector X $\{-1, 1\}^{|V|}$ that minimizes : $\mathcal{L}(X) = \frac{1}{4}\sum_{(i,j)\in E}(X_i - X_j)^2 = \frac{1}{4}X^T L X$ under the constraint $\langle X, \mathbf{1}\rangle = 0$. Here L is the Laplacian matrix of the graph. In other words, we optimize the cut between the partitions under the constraint having a load balanced cut. This is an integer quadratic problem and they are known to

be NP-complete. We first optimize the problem on $\mathbb{R}^{|V|}$ then see how can we return to our original problem. Taking the Lagrangian multipliers we reduce to finding the eigenvector $v \in \mathbb{R}^{|V|}$ of L minimizing

$$min_{v \perp \vec{1}} \frac{v^T L v}{v^T v}$$

Once we find a solution $v \in \mathbb{R}^V$ many methods exist to find the corresponding cut in G. The general method is given using a common procedure called sweep. We look for a threshold value s to help us to form our two partitions :

$$P_1 = \{x \in V | v(x) \leq s\}, P_2 = \{x \in V | v(x) > s\}$$

Many ways to determine this threshold exist :

- Sign cut : The most intuitive one when seeing where we come from. We set $s = 0$, so all $v \in V$ having their corresponding entry in v positive will be put in P_1 and the others in P_2.
- Ratio cut : We choose the value of s that gives the best ratio cut. Cheeger inequality ensures finding a threshold s that gives a cut of ratio at most : $\frac{h(G)^2}{2\Delta}$.
- Bisection cut : Here we want to split the graph into two components of same size, we set s as the median of the $\{v_1, v_2, \ldots, v_n\}$.

As stated in section 3, our case is different since we deal with a graph composed of metanodes on which we have to take into account that many nodes form it.

Let's introduce some notations about the metanode graph G'. We will denote $G' = (V', E')$ the graph associated with its vertices and edges. We will let n' be the number of vertices in G' which corresponds to the number of metanodes. For all $v_i \in V'$, we denote $d_i \in \mathbb{N}^*$ the number of nodes in the original graph forming it. Finally, we denote by Δ_{meta} the maximal degree in G'. Moreover, we associate a weight to the edges between two metanodes being the number of edges crossing the two partitions :

$$\forall (i,j) \in [\![1, n']\!]^2, A_{i,j} = |\delta(v_i, v_j)|$$

A is the adjacency matrix of G', and we will let $L' = N - A$ be the Laplacian matrix of the graph G' where N is the diagonal matrix containing the weighted degree of the vertices in G' on its diagonal.

We will modify the original method, rather than looking for the second eigenvector of L', we will look for the second eigenvector of $D^{-1/2} L' D^{-1/2}$ as in [10]. Here the D matrix contains the information of the number of nodes in each metanode, it will appear naturally when doing the calculation. The reader used to this kind of literature could say this is the definition of the normalized Laplacian, but here our D is the diagonal matrix with the $(d_i)_{i \in [\![1,n]\!]}$ on its diagonal, and not the degree of metanodes.

The metric we will aim to minimize in G' is the following :

$$\phi_{G'}(A') = \frac{\sum_{(v_i,v_j)\in E'}(\mathbf{1}_{v_i\in A'} - \mathbf{1}_{v_j\in A'})^2 A_{i,j}}{\sum_{i=1}^{n'} \mathbf{1}_{v_i\in A'} d_i} = \frac{\mathbf{1}_{A'}^T L' \mathbf{1}_{A'}}{\mathbf{1}_{A'}^T D \mathbf{1}_{A'}}$$

Where we denoted for $A' \subset V'$, $\mathbf{1}_{v_i\in A'}$ the vector with 1 in position v if $v \in A$ and 0 otherwise. If we rewrite $A = \{x \in V | \exists v_i \in A', st\ x \in v_i\}$, we have $\phi_{G'}(A') = \phi_G(A)$.

Now we present the difference between the classical spectral method and the weighted one we use. Algorithm 2 presents the original version that acts on the original non-modified graph.

Algorithm 2 Spectral Bisection

Require: Graph $G = (V, E)$, n, bal. $\in [0, 1]$
$N \leftarrow$ deg. matrix, $A \leftarrow$ adj. matrix
$L \leftarrow N - A$, $f \leftarrow 2^{\text{nd}}$ eigvec of L
$bRC \leftarrow \infty$, $bP \leftarrow \emptyset$
for $t \in \left(\frac{f_i + f_{i+1}}{2}\right)_{i=1}^{n-1}$ **do**
    $S \leftarrow \{i \mid f_i \leq t\}$, $\bar{S} \leftarrow V \setminus S$
    if $\min(|S|, |\bar{S}|) \geq \frac{n}{2} - \text{bal.} \cdot n$ **then** ▷ To ensure the load balancing
        $c \leftarrow \phi_G(S)$
        if $c < bRC$ **then**
            $bRC \leftarrow c$, $bP \leftarrow (S, \bar{S})$
        end if
    end if
end for
return bP

Our modified version which does the spectral algorithm with weights, on the metanode graph is presented in Algorithm 3.

Algorithm 3 Spectral Bisection Weighted

Require: Graph $G' = (V', E')$, n', bal. $\in [0, 1]$
$N \leftarrow$ deg. matrix, $A \leftarrow$ adj. matrix, $D = Diag(d_1, d_2, \ldots, d_{n'})$
$L_{meta} \leftarrow D^{-1/2}(N - A)D^{-1/2}$, $f \leftarrow 2^{\text{nd}}$ eigvec of L_{meta}
$bRC \leftarrow \infty$, $bP \leftarrow \emptyset$
for $t \in \left(\frac{f_i + f_{i+1}}{2}\right)_{i=1}^{n-1}$ **do**
    $S \leftarrow \{i \mid f_i \leq t\}$, $\bar{S} \leftarrow V' \setminus S$
    $X \leftarrow \mathbf{1}_S$
    if $X^\top DX \in [\frac{n}{2} \pm \text{bal.} \cdot n]$ **then** ▷ To ensure the load balancing
        $c \leftarrow \phi_{G'}(S)$
        if $c < bRC$ **then**
            $bRC \leftarrow c$, $bP \leftarrow (S, \bar{S})$
        end if
    end if
end for
return bP

4.2 Analysis of the method

Let λ_2 be the second smallest eigenvalue (0 is the first one) of $L_{meta} = D^{-1/2}(N - A)D^{-1/2}$, i.e. $L_{meta} = D^{-1/2}L'D^{-1/2}$ where and D is the square diagonal matrix with $D(i, i)$ being the number of nodes contained in the metanode i. We suppose our metanodes have size at most M. We can see the analogy between L_{meta} and the normalized Laplacian matrix. The following proof will be mainly based on the fact that the second smallest eigenvalue of L_{meta} verifies the same property than the original Fiedler value. The other ingredient is already given by the analysis made by Spielmann and Teng on the spectral methods in planar graphs [15]. Let's reformulate the classical theorems regarding Fiedler value in our case.

Proposition 3 (Cheeger inequality)**.** *There exists a set $S \subset V'$ of metanodes*[1]*, such that the ratio cut of S is upper bounded : $\phi_{G'}(S) \leq \Delta\sqrt{2\lambda_2 M}$.*

Proof. The classic Cheeger inequality [18] provides us $S \subset V'$ verifying :

$$\frac{\mathbf{1_S^T L' 1_S}}{\mathbf{1}_S^T N \mathbf{1}_S} \leq \sqrt{2\lambda_2 M}$$

M appears in the previous inequality thanks to Horn's inequalities between the spectrum of L' and L_{meta}. Thus we have :

$$\phi_{G'}(S) = \frac{\mathbf{1_S^T L' 1_S}}{\mathbf{1}_S^T D \mathbf{1}_S} = \frac{\mathbf{1_S^T L' 1_S}}{\mathbf{1}_S^T N \mathbf{1}_S} \frac{\mathbf{1_S^T N 1_S}}{\mathbf{1}_S^T D \mathbf{1}_S} \tag{1}$$

$$\leq \sqrt{2M\lambda_2} max_{i \in [\![1,n']\!]} \frac{n_i}{d_i} \leq \sqrt{2M\lambda_2} max_{i \in [\![1,n']\!]} \frac{\phi_G(v_i) d_i}{d_i} \leq \Delta\sqrt{2M\lambda_2} \tag{2}$$

[1]This subset is returned by spectral method we use.

□

We will seek an upper bound to λ_2 so that the ratio cut provided by the spectral algorithm will also be upper bounded. We need the below lemma to obtain a different expression for λ_2, the second smallest eigenvalue of L_{meta}.

Lemme 1. *Let $S \in \mathbb{R}^{n\times n}$ a symmetric positive matrix, such that S admits $d \in \mathbb{R}^n$ as a kernel vector and $dim(Ker\ S) = 1$, then denoting $\lambda_2 \in \mathbb{R}_+^*$ its second smallest eigenvalue (0 is the first one) we have :*

$$\forall k \in \mathbb{N}^*,\ \lambda_2 = min_{U\in\mathbb{R}^{n\times k}, U^T d=\vec{0}} \frac{tr(U^T SU)}{tr(U^T U)}$$

Proof. We fix $k \in \mathbb{N}^*$ and proceed by double inequality, first we deal with the inequality $\leq$. Let $U \in \mathbb{R}^{n\times k}$, and $u_1, u_2, \ldots, u_k$ the columns of U.

$$\frac{tr(U^T SU)}{tr(U^T U)} = \sum_{i=1}^{k} \frac{u_i^T S u_i}{\sum_{i=1}^k \|u_i\|^2} \geq \sum_{i=1}^{k} \frac{\lambda_2 \|u_i\|^2}{\sum_{i=1}^k \|u_i\|^2} = \lambda_2$$

The inequality is justified because u_i is orthogonal to d and by the properties of symmetric positive matrices. Then, let $v \in \mathbb{R}^n$ be the eigenvector associated to λ_2 and V be the matrix whose k columns are all v. We have : $V^T d = \vec{0}$ and

$$\lambda_2 = \sum_{i=1}^{k} \frac{\lambda_2 \|v\|^2}{\sum_{i=1}^k \|v\|^2} = \sum_{i=1}^{k} \frac{v^T S v}{\sum_{i=1}^k \|v\|^2} = \frac{tr(V^T SV)}{tr(V^T V)}$$

which conclude the proof. □

Let's use an other definition of λ_2 so that it will be easier to upper bound it. We will denote $(d_i)_{i\in[\![1,n']\!]} \in \mathbb{R}^n$ the vector containing in $i-$th position the number of nodes in the $i-$th metanode. It verifies : $D\vec{1} = d$.

Proposition 4. *For all $k \in \mathbb{N}^*$, λ_2 the second smallest eigenvalue of $L_{meta} = D^{-1/2}(N - A)D^{-1/2}$ verifies :*

$$\lambda_2 = min_{\vec{v_1},\ldots,\vec{v_{n'}}\in\mathbb{R}^k, \sum d_i\vec{v_i}=\vec{0}} \frac{\sum_{(i,j)\in E_{G'}} \|\vec{v_i} - \vec{v_j}\|^2 A_{i,j}}{\sum_{i\in V'} \|\vec{v_i}\|^2 d_i}$$

Proof. The quantity we need to optimize can be written as :

$$\frac{\sum_{(i,j)\in E_{G'}} \|\vec{v_i} - \vec{v_j}\|^2 A_{i,j}}{\sum_{i\in V} \|\vec{v_i}\|^2 d_i} = \frac{tr(V^T L' V)}{tr(V^T DV)}$$

where $V \in \mathbb{R}^{n'\times k}$ is the matrix whose lines are the $(\vec{v_i})_{i\in[\![1,n']\!]}$ and thus verifies : $V^T d = \vec{0}$. We make a change of variable $U = D^{1/2}V$, so that we minimize :

$$\frac{tr(U^T L_{meta} U)}{tr(U^T U)}$$

for $U \in \mathbb{R}^{n' \times k}$ that verifies $\vec{0} = U^T D^{-1/2} d = U^T D^{-1/2} D\vec{1} = U^T D^{1/2} \vec{1}$. The lemma 2 justifies the first equality, and thus we have the result.

$$\lambda_2 = min_{U \in \mathbb{R}^{n' \times k}, U^T D^{1/2} \vec{1} = \vec{0}} \frac{tr(U^T L_{meta} U)}{tr(U^T U)} \tag{3}$$

$$= min_{V \in \mathbb{R}^{n' \times k}, V^T d = \vec{0}} \frac{tr(V^T L V)}{tr(V^T D V)} \tag{4}$$

$$= min_{\vec{v_1}, ..., \vec{v_{n'}} \in \mathbb{R}^k, \sum d_i \vec{v_i} = \vec{0}} \frac{\sum_{(i,j) \in E_{G'}} \|\vec{v_i} - \vec{v_j}\|^2 A_{i,j}}{\sum_{i \in V'} \|\vec{v_i}\|^2 d_i} \tag{5}$$

□

Now let's state the main result.

Proposition 5. *We have $\lambda_2 \leq M \frac{16\Delta}{n}$. We find by our spectral method a subset $T \subset V'$ verifying $\phi_{G'}(T) \leq M\Delta\sqrt{\frac{32\Delta}{n}}$.*

To show this proposition we follow a similar approach of Spielmann and Teng using Koebe's theorem on planar graphs.

Theorem 1 (Koebe's Theorem)**.** *Let G = (V, E) be a planar graph. Then there exists a set of circles $\{C_1, ..., C_n\}$ in $\mathbb{R}^2$ that are interior disjoint such that circle C_i kisses circle C_j if and only if $(i, j) \in E$.*

This theorem will help us finding an embedding for our planar graphs that will make naturally appear the vectors $(v_i)_{i \in [\![1,n']\!]}$ of proposition 5. Here kisses can be interpreted as touches, meaning that the two disks share exactly one point.

We start using the embedding the above theorem gives us, so we dispose of a plane with disks on it representing the nodes. Then we will use the stereographic projection onto the 3D sphere $\mathbf{S}^2$. Our original disks will be seen as caps onto this sphere. To each of these caps we can then assign a center, cap C_i will be assigned to center $\vec{v_i} \in \mathbf{S}^2$. We denote by p the map that assigns to a cap its center, for example : $p(C_i) = \vec{v_i}$. We denote by $\mathbf{B}_{\mathbb{R}^3}(0, 1)$ the unit centered ball in $\mathbb{R}^3$.

We follow the same steps as in [15] but in the general case where nodes have a weight.

Proposition 6. *There exists an embedding of the graph onto $\mathbf{S}^2$, such that each node v_i corresponds to a cap C_i that kisses an other cap C_j iff $(v_i, v_j) \in E$. Moreover, the $d-$centroid of the cap's center is $\vec{0}$:*

$$\frac{1}{\sum_{i=1}^{n'} d_i} \sum_{i=1}^{n'} d_i p(C_i) = \vec{0}$$

Proof. We consider a family of function defined by:

$$\forall \vec{\alpha} \in \mathbf{B}_{\mathbb{R}^3}(0, 1),\ f_{\vec{\alpha}} : \mathbf{S}^2 \to \mathbf{S}^2 \tag{6}$$

$$z \mapsto \pi_{\vec{\alpha}/\|\vec{\alpha}\|}(D^{1-\|\vec{\alpha}\|}_{\vec{\alpha}/\|\vec{\alpha}\|}(\pi^{-1}_{\vec{\alpha}/\|\vec{\alpha}\|}(z))) \tag{7}$$

where $\pi_{\vec{\alpha}/\|\vec{\alpha}\|}$ is the stereographic projection from the plane tangent to $\mathbf{S}^2$ at $\vec{\alpha}/\|\vec{\alpha}\|$, $\pi^{-1}_{\vec{\alpha}/\|\vec{\alpha}\|}$ is its opposite function that maps a point of the sphere to a point into the plane tangent to $\mathbf{S}^2$ in $\vec{\alpha}/\|\vec{\alpha}\|$. The image of a point z in the hyperplane tangent to $\mathbf{S}^2$ in $\vec{\alpha}/\|\vec{\alpha}\|$ by $\pi_{\vec{\alpha}/\|\vec{\alpha}\|}$ is defined as the intersection of the unit sphere and the line between z and $-\vec{\alpha}/\|\vec{\alpha}\|$. We must be careful that $\pi^{-1}_{\vec{\alpha}/\|\vec{\alpha}\|}$ is not defined everywhere since the point $-\vec{\alpha}/\|\vec{\alpha}\|$ doesn't have an image by $\pi_{\vec{\alpha}/\|\vec{\alpha}\|}$. Moreover we define for $\vec{\alpha}$ in the sphere and $\beta > 0$, $D^{\beta}_{\vec{\alpha}}$ as the dilatation by a factor β of the plane tangent to the sphere at $\vec{\alpha}$. For $\vec{\alpha}$ of norm 1, we extend the definition of $f_{\vec{\alpha}}$ to :

$$f_{\vec{\alpha}}(z) = \begin{cases} -\vec{\alpha} & \text{if } z = -\vec{\alpha} \\ \vec{\alpha} & \text{otherwise} \end{cases}$$

These functions take a point onto the sphere and map it to the plane tangent to $\mathbf{S}^2$ in $\vec{\alpha}/\|\vec{\alpha}\|$. Then a dilatation of this plane is done. After this the point is projected again onto the sphere. One thing to notice is that the more $\|\vec{\alpha}\|$ gets close to 1, the closest the caps will get to $\vec{\alpha}$ after the transformation $f_{\vec{\alpha}}$.

An other thing to notice about this kind of function, when $\|\vec{\alpha}\| \neq 1$, is that they map an interior-disjoint set of caps to an other interior-disjoint set of caps and caps kiss in one set iff they kiss in the other. So if we can find $\vec{\alpha} \in \mathbf{B}_{\mathbb{R}^3}(0,1)$, verifying

$$\sum_{i=1}^{n'} d_i p(f_{\vec{\alpha}}(C_i)) = \vec{0}$$

the result will be given to us. We set for all $i \in [\![1, n']\!]$, $\bar{d}_i = \frac{d_i}{\sum_{j=1}^{n'} d_j}$ and show the equation above for the $(\bar{d}_i)_{i\in[\![1,n']\!]}$ which is a similar statement. To solve some issues of non-continuity for the $\vec{\alpha}$ in $\mathbf{S}^2$, we directly consider for $\varepsilon > 0$, and a cup C :

$$h^{\varepsilon}_{\vec{\alpha}}(C) = \begin{cases} 1 & \text{if } \mathrm{dist}(-\vec{\alpha}, C) > \varepsilon, \\ \frac{\mathrm{dist}(-\vec{\alpha},C)}{\varepsilon} & \text{otherwise.} \end{cases}$$

This function is designed to vanish when $-\vec{\alpha}$ gets too much closer of the cap C. Some issues of discontinuity can happen when $-\vec{\alpha}$ gets into a cap without this term in the next definition :

$$\forall \vec{\alpha} \in \bar{\mathbf{B}}_{\mathbb{R}^3}(0,1), \varphi_{\varepsilon}(\vec{\alpha}) = \sum_{i=1}^{n} h^{\varepsilon}_{\vec{\alpha}}(C_i)\bar{d}_i p(f_{\vec{\alpha}}(C_i)).$$

It is similar to the expression we seek to annihilate. It is a continuous application from $\bar{\mathbf{B}}_{\mathbb{R}^3}(0,1)$ to $\bar{\mathbf{B}}_{\mathbb{R}^3}(0,1)$, thanks to the h^{ε}. Because we have at most 1 point shared by two caps and since the graph has a finite number of nodes, we can choose ε sufficiently small such that at most two caps are concerned by the kind of situation where $h^{\varepsilon}_{\vec{\alpha}}(C) \neq 1$. Since we know that our caps are interior disjoint and we suppose having at least three metanodes, at least one term of the sum will be non null with value some $d_i\vec{\alpha}$.

Then we retrieve ourselves in the same case as in the original proof of [15], so that for all $\vec{\alpha} \in \mathbf{S}^2$, $\varphi(\vec{\alpha})$ belonged to the line between $\vec{0}$ and $\vec{\alpha}$, being closer to $\vec{\alpha}$ than $-\vec{\alpha}$.

As a consequence of Brouwer's fixed point theorem we dispose of $\vec{\alpha_0}^{\varepsilon} \in \mathbf{B}_{\mathbb{R}^3}(0,1)$ such that : $\varphi_\varepsilon(\vec{\alpha_0}^{\varepsilon}) = \vec{0}$. We want to show that $\vec{\alpha_0}^{\varepsilon}$ can be taken inside the unit ball of radius $1-\epsilon$ so that all h_ε contributions equal 1.

We choose $\varepsilon' > 0$ such that for all $\vec{\alpha}$ of norm greater than $1-\varepsilon'$ we have at least $n'-2$ caps $(f_{\vec{\alpha}}(C_1), f_{\vec{\alpha}}(C_2), \ldots, f_{\vec{\alpha}}(C_{n'-2}))$ include in $\mathbf{B}(\vec{\alpha}, \varepsilon')$. It exists thanks to the action of contraction of $f_{\vec{\alpha}}$.

Now we need one more assumption that differs from the original proof : $\forall (i,j) \in [\![1,n']\!]^2,\ d_i + d_j < n/2$. This is a reasonable assumption in the case of metanodes graph, it means that two metanodes will never cover more than the half of the graph. This assumption ensures us that for $\vec{\alpha}$ of norm greater than $1-\varepsilon'$ the expression doesn't annihilate. We then get the result by the previous discussion.

□

Now we dispose of an efficient embedding that will help us upper bounding the fraction of Proposition 5.

Proof of Proposition 6. By the previous discussion, we dispose of an embedding of the graph into the 3D-sphere $\mathbf{S}^2$ such that each node corresponds to the center of a cup $\vec{v_i}$. We denote c_i the corresponding cup and its radius cup by r_i. We choose the embedding such that : $\sum d_i \vec{v_i} = \vec{0}$ as Proposition 7 provides us. We seek an upper bound on the right quantity in Proposition 5.

Let's deal with the numerator first :

$$\sum_{(i,j)\in E_{G'}} \|\vec{v_i} - \vec{v_j}\|^2 A_{i,j} \leq 2 \sum_{i=1}^{n} \sum_{j=1}^{n'} (r_i^2 + r_j^2) A_{i,j} = 4 \sum_{i=1}^{n'} r_i^2 \sum_{j=1}^{n'} A_{i,j}$$

This is given by the fact that the distance between two vertices $\|\vec{v_i} - \vec{v_j}\|^2$ is at most $(r_i + r_j)^2$, by a geometric argument. Moreover since cluster i have size d_i, we have : $\sum_{j=1}^{n'} A_{i,j} \leq \Delta M$. It is because $\sum_{j=1}^{n} A_{i,j}$ corresponds to the number of external connections with metanode i and each node in G have at most Δ neighbors.

Since the caps are interior disjoint and lies on the sphere, we have by comparing surfaces : $\sum_{i=1}^{n'} \pi r_i^2 \leq 4\pi$. Finally,

$$\sum_{(i,j)\in E_{G'}} \|\vec{v_i} - \vec{v_j}\|^2 A_{i,j} \leq 4\Delta M \sum_{i=1}^{n'} r_i^2 \leq 16 \Delta M$$

For the denumerator, since the $(\vec{v_i})_{i\in[\![1,n']\!]}$'s lie on the unit sphere, we have

$$\sum_{i=1}^{n'} \|\vec{v_i}\|^2 d_i = \sum_{i=1}^{n'} d_i = n$$

Graph	nsubdoms	Spectral Bisec.	CSB 1	CSB 2	CSB 3	CSB 4	CSB 5
Cantilever Bean	2	20	20	22	26	26	26
Cantilever Bean	4	60	60	70	70	76	92
Cantilever Bean	8	140	142	156	180	212	282
Chamfer 5.3	2	96	94	102	110	116	132
Chamfer 5.3	4	268	292	308	326	348	390
Chamfer 5.3	8	538	566	576	654	730	706

Table 1: Interface Size (IS) between partitions.

We finally get thanks to Proposition 3, the inequality :

$$\lambda_2 \leq \frac{16\Delta M}{n}$$

Thanks to proposition 4: the spectral method gives a subset T of V' that verifies $\phi_{G'}(T) \leq M\Delta\sqrt{\frac{32\Delta}{n}}$. □

5 Numerical experiments

In this section we compare the performances of our coarsening algorithm with the original spectral recursive bisection algorithm. We seek to have almost load balance partitions so we guarantee a minimum load balancing, choosing a threshold value of our sweep procedure ensuring a 0.05 load balancing. We denote in the tables by CSB the Coarse Spectral Bisection algorithm, with a number corresponding to the level of coarsification done.

We consider two different test cases, namely a cantilever bean of size 10×100, meshed with quadrangle elements, and a chamfered meshed with triangle elements. Figure 1 shows an example of the partitioning obtained with the proposed method. Table 1 and Table 2 shows the interface size and the load balancing ratio, confirming the theoretical analysis of the proposed method. Table 3 shows the computational time, which also confirm the faster execution of our method for the different geometries.

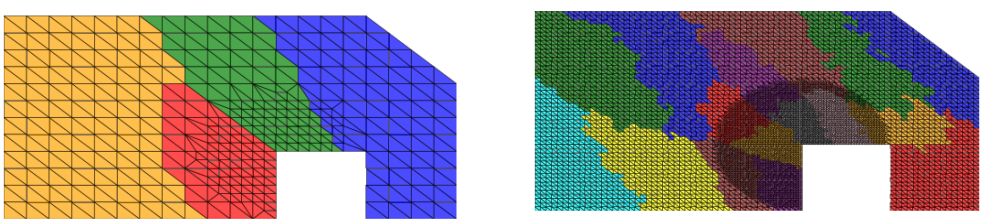

Figure 1: Partitioning of the Chamfer into four partitions (left) and sixteen partitions (right).

Graph	nsubdoms	Spectral Bisec.	CSB 1	CSB 2	CSB 3	CSB 4	CSB 5
Cantilever Bean	2	1.00	1.04	1.04	1.17	1.03	1.04
Cantilever Bean	4	1.00	1.08	1.22	1.21	1.09	1.26
Cantilever Bean	8	1.08	1.13	1.36	1.22	1.29	1.27
Chamfer 5.3	2	1.04	1.16	1.17	1.17	1.18	1.20
Chamfer 5.3	4	1.04	1.22	1.21	1.25	1.20	1.24
Chamfer 5.3	8	1.06	1.36	1.46	1.29	1.43	1.42

Table 2: Ratio of the number of elements in the largest partition over the smallest one.

Graph	nsubdoms	Spectral Bisec.	CSB 1	CSB 2	CSB 3	CSB 4	CSB 5
Cantilever Bean	2	0.92	0.59	0.62	0.65	0.65	0.70
Cantilever Bean	4	0.92	0.73	0.67	0.69	0.70	0.67
Cantilever Bean	8	1.00	0.72	0.71	0.74	0.74	0.69
Chamfer 5.3	2	649.00	140.34	76.73	63.68	61.62	61.00
Chamfer 5.3	4	980.00	178.40	85.70	65.40	63.80	62.50
Chamfer 5.3	8	887.67	240.00	105.00	66.00	62.00	64.00

Table 3: Computational time in seconds.

6 Concluding remarks

After proposing a new method for mesh partitioning and its associated algorithm, we prove some theoretical properties in the case of planar meshes. The numerical experiments confirm that, compared to the original spectral recursive method, our method is always faster from a computational point of view, despite a small modification of the load balancing. This property is mainly due to the vertex contraction occuring during the matching edges process. Further investigations are under current analysis in order to extend these results to non-planar meshes.

Acknowledgements

The authors acknowledge partial financial support from the Urba(IA) project funded by the DIAT demonstrator for frugal Artificial Intelligence for territory.

References

[1] K. Andreev and H. Racke, *Balanced graph partitioning*. Theory of Computing Systems, vol. 39, no. 6, pp. 929–939, 2006

[2] S.T. Barnard and H.D. Simon. *A Fast Multilevel Implementation of Recursive Spectral Bisection for Partitioning Unstructured Problems.* In 6th SIAM Conference on Parallel Processing for Scientific Computing (PPSC). 711–718. 1993.

[3] B. Delaunay. *Sur la sphère vide*, Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk, vol. 7, pp. 793–800, 1934.

[4] R. Diestel. *Graph Theory*, Springer, Sixth edition 2025

[5] A. El Kerim, P. Gosselet, and F. Magoulès. *Asynchronous global–local non-invasive coupling for nonlinear monotone patches : Application to plasticity problems.* Computer Methods in Applied Mechanics and Engineering, 430(117166), 2024.

[6] B. Faucard, A.G. Sorguc, F. Magoulès, and I. Hagiwara. *Refining technique for multilevel graph k-partitioning and its application on domain decomposition non overlapping Schwarz technique for urban acoustic pollution.* Transaction of the Japan Society for Simulation Technology, 1(2) :17–27, 2009.

[7] C. Farhat, N. Maman, and G. Brown. *Mesh partitioning for implicit computations via iterative domain decomposition.* Int. J. Num. Meth. Engrg., 38:989–1000, 1995.

[8] G. Gbikpi-Benissan and F. Magoulès. *Accurate implementation of two-level asynchronous domain decomposition solvers.* Advances in Engineering Software, 193(103660), 2024

[9] G. Gbikpi-Benissan and F. Magoulès. *Asynchronous multisplitting-based primal schur method.* Journal of Computational and Applied Mathematics, 425(115060), 2023.

[10] Y. Jin, A. Loukas, and J.F. JaJa. *Graph Coarsening with Preserved Spectral Properties*. Proceedings of Machine Learning Research, Chiappa, Silvia and Calandra, Roberto. PLMR 4452-4462.

[11] G. Karypis, and V. Kumar. *A fast and high quality multilevel scheme for partitioning irregular graphs.* SIAM J. on Scientific Computing, 1998, 359-392.

[12] F. Magoulès, P. Iványi, and B.H.V. Topping. *Non-overlapping Schwarz methods with optimized transmission conditions for the Helmholtz equation*. Computer Methods in Applied Mechanics and Engineering, 193(45–47) :4797–4818, 2004.

[13] F. Magoulès, P. Iványi, and B.H.V. Topping. *Convergence analysis of Schwarz methods without overlap for the Helmholtz equation.* Computers and Structures, 82(22) :1835–1847, 2004.

[14] F.P. Preparata and M.I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.

[15] D.A. Spielman and S.-H. Teng. *Spectral partitioning works: planar graphs and finite element meshes*, Proceedings of 37th Conference on Foundations of Computer Science, Burlington, VT, USA, 1996, pp.96-105,

[16] T. Tsuji, F. Magoulès, K. Uchida, and T. Oyama. *A partitioning technique for a waveform relaxation method using eigenvectors in the transient stability analysis of power systems.* Power Systems, IEEE Transactions on, 30(6) :2867–2879, 2015.

[17] C. Walshaw and M. Cross. Mesh Partitioning Techniques and Domain Decom-

position Techniques, pages 27-58, Civil-Comp Ltd., 2007.
[18] Four Cheeger-type Inequalities for Graph Partitioning Algorithms, Fan ICCM 2007 · Vol. II · 1–4