

# **RAPPORT FINAL DU PSC MAT03 : OPTIMISATION SPECTRALE SUR LES GRAPHES QUANTIQUES**

30 avril 2020

Khatouri Zouhair, Paulo Elpidio Alves Sampaio, Chen Xi,  
Fafe Soufiane, Anon Daniel Wilfried Hassan



## TABLE DES MATIÈRES

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Approche théorique du problème</b>	<b>3</b>
2.1	Définitions préliminaires . . . . .	3
2.1.1	Définitions et notations . . . . .	3
2.1.2	Graphes métriques . . . . .	4
2.1.3	Graphes quantiques . . . . .	6
2.2	Conditions aux sommets . . . . .	7
2.2.1	Conditions aux sommets et notations . . . . .	7
2.2.2	Conditions aux sommets et Hamiltonien autoadjoint . . . . .	8
2.2.3	Conditions de Neumann-Kirchhoff . . . . .	9
2.3	Exemples par calcul direct . . . . .	10
2.3.1	Un graphe trivial - un intervalle . . . . .	10
2.3.2	Le graphe mandarine équilatéral à $n$ arêtes . . . . .	11
2.4	Matrices de diffusion et équation séculaire . . . . .	12
2.4.1	Graphes quantiques et matrices de diffusion . . . . .	12
2.4.2	Matrice de diffusion des liaisons et équation séculaire . . . . .	13
2.4.3	Opérateurs du premier ordre et matrices de diffusion . . . . .	14
2.5	Spectre de certains graphes à partir de l'équation séculaire . . . . .	16
2.5.1	Graphe mandarine à $n$ arêtes . . . . .	16
2.5.2	Graphe étoile à $n$ arêtes . . . . .	17
2.6	Propriétés spectrales d'un graphe quantique . . . . .	18
2.7	Les formules de la trace . . . . .	21
<b>3</b>	<b>Approche Numérique du problème</b>	<b>23</b>
3.1	Méthode variationnelle et éléments finis . . . . .	23
3.2	Méthode de la résolution de l'équation séculaire . . . . .	26
3.2.1	Description de la méthode . . . . .	27
3.2.2	Tests du code sur des graphes "simples" . . . . .	27
3.2.3	Traitement du cas des graphes à boucles ou à arêtes multiples . . . . .	29
3.2.4	Utilisation de l'outil numérique pour des problèmes ouverts . . . . .	31
3.2.5	Exploration numérique de cas plus généraux . . . . .	35
3.3	Méthode probabiliste : recuit simulé . . . . .	38
<b>4</b>	<b>Appendice</b>	<b>41</b>
4.1	La classe Graph . . . . .	41
4.2	La classe GraphFunction . . . . .	45
4.3	Le main . . . . .	48
4.4	Le main du recuit simulé . . . . .	51

# 1

## INTRODUCTION

L'étude d'opérateurs de type Schrödinger sur des graphes métriques est un domaine en expansion de la physique mathématique, qui est motivé à la fois par l'application directe des modèles de graphes à des phénomènes physiques, et par l'utilisation de graphes comme un cadre plus simple permettant d'étudier les phénomènes complexes de la mécanique quantique.

Les graphes quantiques ont été étudiés pour la première fois par Linus Pauling en tant que modèles d'électrons libres dans des molécules organiques dans les années trente (Fig. 1). Ils apparaissent également dans divers contextes mathématiques, par exemple, les modèles de la théorie du chaos quantique, dans l'étude des guides d'ondes, dans les cristaux photoniques et dans la localisation d'Anderson, ou encore comme limite pour le rétrécissement de fils minces. Les graphes quantiques sont devenus des modèles de premier plan en physique mésoscopique, utilisés pour obtenir une compréhension théorique de la nanotechnologie.

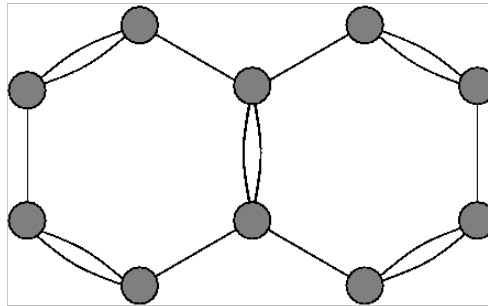


FIGURE 1 – Graphe relatif à la molécule de naphthalène, l'un des responsables de l'émergence de la théorie des graphes quantiques

# 2

## APPROCHE THÉORIQUE DU PROBLÈME

### 2.1 DÉFINITIONS PRÉLIMINAIRES

On commence par introduire quelques notions élémentaires de la théorie graphes, ainsi que les notations utilisées dans la suite du rapport.

#### 2.1.1 • DÉFINITIONS ET NOTATIONS

Un graphe  $\Gamma$  consiste en un ensemble fini de sommets  $V$ , et un ensemble fini  $E$  d'arêtes reliant les sommets. Le degré  $d_u$  d'un sommet  $u$  est le nombre d'arêtes qui en émanent.

Parfois, il faut prendre en compte l'orientation des arêtes. Un tel graphe est dit être orienté. Cela signifie que chaque arête a un sommet origine et un sommet terminal. Les arêtes orientées seront appelées arcs, et l'ensemble de toutes les arcs est noté  $A$ . Le sommet origine et le sommet terminal d'un arc sont spécifiés via les fonctions  $o : A \rightarrow V$  et  $t : A \rightarrow V$ , c'est-à-dire qu'un arc  $b$  commence au sommet  $o(b)$  et se termine à  $t(b)$ .

On peut passer d'un graphe non orienté à un graphe orienté en affectant deux arcs  $b$  et  $\bar{b}$  avec des directions opposées à chaque arête  $e$ . Un exemple d'une telle procédure est donné dans la figure 2. Soit  $\Gamma$  un graphe non

orienté, on désigne par  $\tilde{\Gamma}$  le graphe orienté obtenu par la procédure de duplication qu'on vient juste de décrire. L'ensemble des arcs de  $\Gamma$  est symétrique dans le sens où  $b \in A$  si et seulement s'il existe un autre arc  $\bar{b} \in A$  tel que  $o(b) = t(\bar{b})$  et  $t(b) = o(\bar{b})$ . L'arc  $\bar{b}$  est appelé le renversement de  $b$ . L'application "renversement" est idempotente :  $\bar{\bar{b}} = b$ .

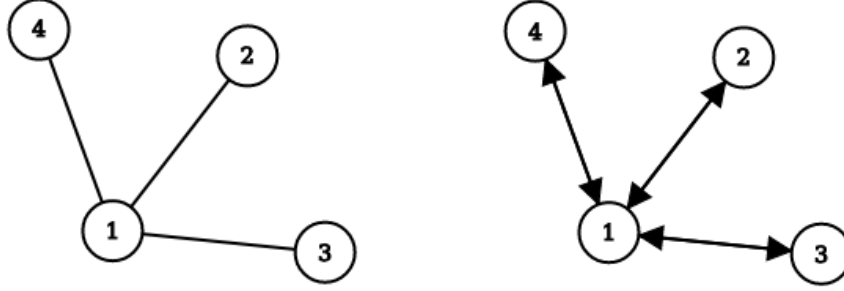


FIGURE 2 – Un graphe non orienté  $\Gamma$  (à gauche), et le graphe orienté correspondant  $\tilde{\Gamma}$  obtenu en dupliquant des arcs (à droite).

Cette spécification de graphes, qui est étudiée en théorie des graphes et en combinatoire, traite les arêtes simplement comme une relation sur l'ensemble des sommets. On qualifiera ce type de graphes de discret ou combinatoire.

On va maintenant introduire une nouvelle spécification pour les graphes, où on considérera les arêtes comme des segments unidimensionnels. On va associer à ce type de graphes des structures supplémentaires qui en feront d'eux des graphes métriques ou quantiques comme on va le voir.

### 2.1.2 • GRAPHES MÉTRIQUES

**Définition 2.1.** (*Graphes métriques*) Un graphe  $\Gamma = (\mathcal{V}, \mathcal{E})$  est dit métrique si :

1. À chaque arc  $b \in \mathcal{E}$  se on attribue une longueur positive  $L_b \in (0, \infty)$ .
2. Les longueurs des arcs qui sont des renversements les unes des autres sont égales :  $L_b = L_{\bar{b}}$ . Ainsi peut-on définir la longueur d'une arête non orientée  $e$ , notée  $L_e$ .
3. À chaque arc on attribue une coordonnée  $x_b \in [0, L_b]$  croissant dans la direction de cet arc.
4. La relation  $x_{\bar{b}} = L_b - x_b$  est vraie pour tout arc  $b \in \mathcal{E}$ .

Si les longueurs de tous les arcs d'un graphe métrique sont égales, ce graphe est dit équilatéral.

On se remarquera que les points d'un graphe métrique  $\Gamma$  ne sont pas seulement ses sommets, mais aussi tous les points intermédiaires sur les arcs. Ainsi, lorsque on parle des fonctions définies sur  $\Gamma$ , on les considère comme définies sur chacun de ses arcs.

La notion de continuité s'étend naturellement à l'ensemble des fonctions définies sur un graphe métrique, i.e. les restrictions à chaque arc du graphe sont continues et les valeurs de ces restrictions coïncident sur les sommets communs. On définit ainsi l'espace des fonctions continues  $C(\Gamma)$ . Chaque arc étant isomorphe à un intervalle réel, on a donc une définition naturelle de la mesure de Lebesgue sur un graphe métrique, qui va nous servir dans la définition d'autres espaces fonctionnels sur les graphes métriques.

### Les espaces de Sobolev

Ici, on fera quelques rappels et définitions sur certains espaces de fonctions. On définit d'abord les espaces de Sobolev, qui constituent le contexte traditionnel de résolution des équations différentielles.

Soit  $\Omega$  un ouvert de  $\mathbb{R}^n$ . On connaît déjà les espaces  $C^m(\Omega)$ , où  $m$  est un entier positif. La notion de fonction de classe  $C^m$  est évidemment fondamentale, de sorte que les espaces  $C^m(\Omega)$  soient des objets naturels. Cependant, ces espaces ne sont pas très commodes pour les problèmes que nous allons traiter ; en effet, ces derniers utilisent de manière naturelle des normes intégrales basées sur  $L^2$  et  $C^m(\Omega)$  n'est un espace de Banach avec aucune d'entre elles.

Les espaces de Sobolev constituent un compromis entre les espaces  $C^m(\Omega)$  et les espaces de Lebesgue  $L^p(X)$ , au sens suivant :

1. Ces espaces sont naturellement des espaces de Banach, et même de Hilbert.
2. Ces espaces se plongent dans des espaces  $C^m(\Omega)$ .

**Définition 2.2.** (*Espaces de Sobolev*) Soient  $\Omega$  un ouvert de  $\mathbb{R}^n$ , un entier  $k \in \mathbb{N}$  et  $p \in [1, \infty]$ . L'espace de Sobolev  $W^{k,p}(\Omega)$  est défini par :

$$W^{k,p}(\Omega) = \{f \in L^p(\Omega) \mid \partial^\alpha f \in L^p(\Omega) \text{ pour } |\alpha| \leq k\}$$

Où  $\partial$  représente la dérivée faible. Cet espace est muni de la norme

$$\|f\|_{W^{k,p}(\Omega)} = \sum_{|\alpha| \leq k} \|\partial^\alpha f\|_{L^p(\Omega)}$$

où on peut montrer que les espaces  $W^{k,p}(\Omega)$  sont des espaces de Banach.

Un cas particulier très important, avec lequel nous travaillerons dans ce rapport est celui des espaces de Sobolev où  $p = 2$ . On note, pour tout  $k \in \mathbb{N}$ ,

$$H^k(\Omega) := W^{k,2}(\Omega)$$

En plus des propriétés des espaces  $W^{k,p}(\Omega)$ , les espaces  $H^k(\Omega)$  possèdent une propriété supplémentaire : ce sont des espaces de Hilbert, pour le produit scalaire :

$$\langle f, g \rangle_{H^k(\Omega)} = \sum_{|\alpha| \leq k} \langle \partial^\alpha f, \partial^\alpha g \rangle_{L^2(\Omega)}$$

Et on rappelle que :

$$\langle \phi, \psi \rangle_{L^2(\Omega)} = \int_{\Omega} \overline{\phi(x)} \psi(x) dx$$

Lorsqu'on est face à une équation aux dérivées partielles, il est courant de se donner des conditions aux bords, par exemple  $f|_{\Sigma} = 0$ , où  $\Sigma$  est un hyperplan de  $\mathbb{R}^n$ . Evidemment, pour tout  $f \in C^m(\mathbb{R}^n)$  et pour tout hyperplan  $\Sigma$  de  $\mathbb{R}^n$ , la restriction  $f|_{\Sigma}$  est une fonction de classe  $C^m$  sur  $\Sigma$ .

Cette restriction n'est pas a priori bien définie dans le cadre des espaces de Sobolev, puisque  $H^k(\mathbb{R}^n)$  est un sous-espace de  $L^2(\mathbb{R}^n)$ , de sorte qu'une fonction de  $H^k(\mathbb{R}^n)$  est une classe d'équivalence de fonctions mesurables qui coïncident sur le complémentaire d'un ensemble de mesure nulle. Comme tout hyperplan de  $\mathbb{R}^n$  est de mesure nulle, on ne peut donc pas parler de la restriction à un hyperplan d'une fonction de  $H^k(\mathbb{R}^n)$ .

On a toutefois un résultat positif dans cette direction, connu sous le nom de "Théorème de Trace" :

Soit  $\Omega$  un ouvert borné de  $\mathbb{R}^n$ . Considérons l'opérateur linéaire :

$$T : C^1(\overline{\Omega}) \rightarrow L^p(\partial\Omega)$$

où  $\overline{\Omega}$  est l'adhérence de  $\Omega$ , et  $\partial\Omega$  est son bord, vérifiant donc :

$$Tu = u|_{\partial\Omega}$$

Le domaine de  $T$  est un sous-ensemble de l'espace de Sobolev  $W^{1,p}(\Omega)$ . Par ailleurs, il existe une constante  $C$ , dépendant uniquement de  $\Omega$  et  $p$ , telle que :

$$\forall u \in C^1(\overline{\Omega}), \|Tu\|_{L^p(\partial\Omega)} \leq C\|u\|_{W^{1,p}(\Omega)}$$

Alors, comme les fonctions  $C^1(\Omega)$  sont denses dans  $W^{1,p}(\Omega)$ , l'opérateur  $T$  admet un unique prolongement continu (qui est donc linéaire) :

$$T : W^{1,p}(\Omega) \rightarrow L^p(\partial\Omega)$$

défini sur l'espace entier  $W^{1,p}(\Omega)$ .  $T$  est appelé opérateur trace. La restriction (ou trace)  $u|_{\partial\Omega}$  d'une fonction  $u$  de  $W^{1,p}(\Omega)$  est alors donnée par l'action de  $T$  sur  $u$ .

Pour  $p = 2$ , on prouve que les classes  $f \in H^2(\Omega)$  admettent des représentants continus  $\partial\Omega$ , ainsi que leurs premières dérivées<sup>1</sup>. Cependant, les dérivées d'ordre supérieur n'ont pas de tels représentants. Ce résultat sera utilisé plus tard dans la définition des conditions aux sommets pour les graphes métriques.

## Espaces standard de fonctions dans $\Gamma$

Nous sommes maintenant prêts à adapter ces espaces de fonctions standard au contexte des graphes métriques.

### Définition 2.3.

1. L'espace  $L^2(\Gamma)$  sur  $\Gamma$  est constitué de fonctions mesurables, et de carré intégrables sur chaque arête  $e$ , et telles que :

$$\|f\|_{L^2(\Gamma)}^2 := \sum_{e \in \mathcal{E}} \|f\|_{L^2(e)}^2$$

2. L'espace de Sobolev  $H^1(\Gamma)$  se compose de toutes les fonctions continues sur  $\Gamma$ , qui appartiennent à  $H^1(e)$  pour chaque arête  $e$ , muni de la norme

$$\|f\|_{H^1(\Gamma)}^2 := \sum_{e \in \mathcal{E}} \|f\|_{H^1(e)}^2$$

On note que la condition de continuité imposée dans l'espace de Sobolev  $H^1(\Gamma)$  implique que toute fonction  $f$  de cet espace prend la même valeur à un sommet sur toutes les arêtes qui lui sont adjacentes, et donc,  $f(v)$  est déterminée d'une façon univoque.

Les espaces de Sobolev  $H^k(\Gamma)$  d'ordre  $k$  supérieur à 1 ne semblent pas avoir de définitions naturelles dans le cadre des graphes métriques, car contrairement au cas unidimensionnel, il n'y a pas de conditions naturelles que ces fonctions doivent satisfaire aux sommets. Cependant, on utilisera souvent des espaces où la régularité est imposée uniquement le long des arcs, sans aucune condition de jonction aux sommets. On introduira la notation qui place un tilde en haut de l'espace pour signifier ceci :

**Définition 2.4.** On note  $\tilde{H}^k(\Gamma)$  l'espace :

$$\tilde{H}^k(\Gamma) := \bigoplus_{e \in \mathcal{E}} H^k(e)$$

C'est l'espace des fonctions  $f$  sur  $\Gamma$ , dont la restriction à chaque arête  $e$  appartient à l'espace de Sobolev  $H_k(e)$ , muni de la norme

$$\|f\|_{\tilde{H}^k(\Gamma)}^2 := \sum_{e \in \mathcal{E}} \|f\|_{H^k(e)}^2$$

### 2.1.3 • GRAPHES QUANTIQUES

Une fois qu'on a défini la notion d'un graphe métrique, et les espaces de fonctions sur ce type de graphe, on va maintenant faire agir des opérateurs différentiels sur ces espaces, ce qui donnera naissance à une nouvelle structure mathématique, dite graphe quantique.

---

1. Ces affirmations ne sont vraies que dans le cas unidimensionnel, mais cela ne pose pas de problème, car les graphes quantiques sont de dimension 1.

**Définition** Un graphe quantique est un graphe métrique équipé d'un opérateur différentiel  $\mathcal{H}$ , souvent appelé hamiltonien, qui, en imposant des conditions aux sommets appropriés aux fonctions de son domaine, est autoadjoint. Autrement dit, un graphe quantique est un triplet : {graphe métrique  $\Gamma$ , hamiltonien  $\mathcal{H}$ , conditions aux sommets}

L'hamiltonien qu'on considère le plus souvent est l'opérateur qui agit comme l'opposé de la dérivée seconde sur chaque arête du graphe :

$$f \mapsto -\frac{d^2 f}{dx^2}$$

Où  $x$  est l'abscisse le long d'une arête quelconque.

On peut également considérer des opérateurs plus généraux, comme par exemple l'opérateur de Schrödinger :

$$\mathcal{H}(f)(x) := -\frac{d^2 f}{dx^2}(x) + V(x)f(x)$$

Notez que pour ces deux opérateurs, la direction de l'arête est sans importance. Ce n'est plus le cas si l'on veut inclure des termes avec les dérivés de premier ordre ; dans ces cas, on doit supposer que le graphe est orienté, et considérer des arcs plutôt que des arêtes.

Pour un potentiel "décent"  $V$ , l'exigence de régularité naturel issue de la théorie des équations différentielles ordinaires est que  $f$  appartient à l'espace de Sobolev  $H^2(e)$  sur chaque arête  $e$ . Des conditions aux sommets appropriées doivent encore être ajoutées : c'est l'objet de la section suivante.

## 2.2 CONDITIONS AUX SOMMETS

Lorsqu'on considère des graphes quantiques, on souhaite que l'hamiltonien soit autoadjoint. Une telle propriété ne dépend pas uniquement de l'opérateur lui-même mais aussi, comme on va le voir dans cette section, des conditions aux sommets utilisées.

### 2.2.1 • CONDITIONS AUX SOMMETS ET NOTATIONS

On convient que les fonctions du domaine de l'opérateur hamiltonien appartiennent à l'espace de Sobolev  $H^2(e)$  sur chaque arête  $e$ , c'est-à-dire qu'on ne considère que des fonctions de l'espace  $\tilde{H}^2(\Gamma)$ . On a vu que dans ce cas, par le "théorème de trace", que  $f \in H^2(e)$  et ses dérivées premières ont des valeurs bien définies aux extrémités de l'arête  $e$ . Par conséquent, les conditions aux sommets peuvent impliquer uniquement les valeurs de  $f$  et  $df/dx$  à un sommet  $v$ . On pourrait permettre aux conditions aux sommets d'impliquer les valeurs à différents sommets. Par exemple, la condition de périodicité pour une fonction  $f$  sur le segment  $[0, 1]$  :  $f(0) = f(1)$  implique les deux sommets 0 et 1. Cependant, on ne traitera que ce qu'on appelle "conditions locales aux sommets", c'est-à-dire, le type de conditions aux sommets qui impliquent les valeurs des fonctions et de leurs dérivées à un seul sommet à la fois. Un graphe arbitraire ressemble localement à un graphe en étoile.

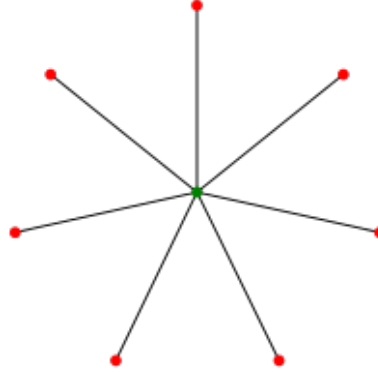


FIGURE 3 – Localement autour d'un sommet, chaque graphe ressemble à un graphe en étoile

Introduisons maintenant quelques notations. Soit  $f \in \tilde{H}^2(\Gamma)$  une fonction sur le graphe  $\Gamma$  définie le long des  $d$  arêtes du graphe en étoile qui constitue le voisinage d'un sommet  $v$ . On note  $F(v)$  le vecteur colonne définie par :

$$F(v) := (f_1(v), \dots, f_d(v))^t = \begin{pmatrix} f_1(v) \\ \vdots \\ f_d(v) \end{pmatrix}$$

des valeurs prises par la fonction  $f$  au sommet  $v$  (si  $f$  est continue au sommet  $v$ , alors ces valeurs sont identiques).

De façon analogue,

$$F'(v) := (f'_1(v), \dots, f'_d(v))^t = \begin{pmatrix} f'_1(v) \\ \vdots \\ f'_d(v) \end{pmatrix}$$

est le vecteur colonne des valeurs au sommet  $v$  des dérivées de  $f$  le long de ces arêtes dans les directions sortantes.

Puisque  $\mathcal{H}$  agit comme un opérateur de second ordre sur chaque arête, il faut établir deux conditions par arête. Par conséquent, à chaque sommet  $v$ , le nombre de conditions coïncide avec le degré  $d_v$  du sommet. Étant donné que pour les fonctions de  $H^2$ , les conditions sur chaque arête peuvent impliquer uniquement les valeurs de la fonction et de sa dérivée aux extrémités, la forme la plus générale d'une telle condition linéaire est la suivante :

$$A_v F(v) + B_v F'(v) = 0$$

Ici,  $A_v$  et  $B_v$  sont des matrices de taille  $(d_v \times d_v)$ . Afin de garantir que le nombre correct de conditions indépendantes (égal au degré  $d_v$  du sommet) est imposé, le rang de la matrice  $(A_v B_v)$ , de taille  $(d_v \times 2d_v)$  doit être égal à  $d_v$ , c'est-à-dire maximal. Cependant, cette condition n'est pas suffisante pour garantir que le hamiltonien est autoadjoint. Dans la section suivante, on discutera en détail des conditions nécessaires et suffisantes pour cela.

## 2.2.2 • CONDITIONS AUX SOMMETS ET HAMILTONIEN AUTOADJOINT

Désormais, on se limite à l'étude de l'opérateur hamiltonien couramment utilisé  $\mathcal{H}$ , à savoir celui qui agit comme  $-\frac{d^2}{dx^2}$  sur chaque arête  $e$ .

Le théorème suivant fournit différentes descriptions équivalentes de toutes les conditions aux sommets rendant  $\mathcal{H}$  autoadjoint.



**Théorème 2.5.** Soit  $\Gamma$  un graphe métrique et  $\mathcal{H}$  l'opérateur agissant comme  $-\frac{d^2}{dx_e^2}$  sur chaque arête de  $\Gamma$ , sur le domaine  $\tilde{H}^2(\Gamma)$ . Cet opérateur est autoadjoint si et seulement si les conditions aux sommets peuvent être écrites sous l'une des trois formes suivantes :

1. Pour chaque sommet  $v$  de degré  $d_v$ , il existe des matrices  $A_v$  et  $B_v$  de taille  $(d_v \times d_v)$  telles que :
  - (a) La matrice  $(A_v B_v)$ , de taille  $d_v \times 2d_v$  a un rang maximal.
  - (b) La matrice  $A_v B_v^*$  est autoadjointe.
  - (c) Les valeurs aux bords de  $f$  satisfont :

$$A_v F(v) + B_v F'(v) = 0$$

2. Pour chaque sommet  $v$  de degré  $d_v$ , il existe une matrice unitaire  $U_v$ , telle que les valeurs aux bords de  $f$  vérifient :

$$i(U_v - \mathbb{I})F(v) + (U_v + \mathbb{I})F'(v) = 0$$

où  $\mathbb{I}$  est la matrice identité  $d_v \times d_v$ .

3. Pour chaque sommet  $v$  de degré  $d_v$ , il existe trois projecteurs orthogonaux (et mutuellement orthogonaux)  $P_{D,v}$ ,  $P_{N,v}$  et  $P_{R,v} := \mathbb{I} - P_{D,v} - P_{N,v}$  agissant sur  $\mathbb{C}^{d_v}$ , et un opérateur auto-adjoint inversible  $\Lambda_v$  agissant sur le sous-espace  $P_{R,v}\mathbb{C}^{d_v}$ , de sorte que les valeurs aux bords de  $f$  vérifient :

$$\begin{cases} P_{D,v}F(v) = 0 & \text{Partie Dirichlet} \\ P_{N,v}F'(v) = 0 & \text{Partie Neumann} \\ P_{R,v}F'(v) = \Lambda_v P_{R,v}F(v) & \text{Partie Robin} \end{cases}$$

La démonstration de ce théorème est trop technique et sort du cadre de ce rapport, dont le but est de se concentrer sur les exemples d'application de ces théorèmes.

### 2.2.3 • CONDITIONS DE NEUMANN-KIRCHHOFF

Après avoir discuté des conditions générales aux sommets, ici, on traitera un exemple couramment utilisé, nommé les conditions "Neumann", "Kirchhoff" ou "standard", à savoir :

$$\begin{cases} f \text{ est continue sur } \Gamma \\ \text{En chaque sommet } v \text{ on a } \sum_{e \in E_v} \frac{df}{dx_e}(v) = 0 \end{cases}$$

Ici, la somme est prise sur l'ensemble  $E_v$  de toutes les arêtes  $e$  incidentes au sommet  $v$ , et les dérivées sont supposées être prises dans les directions qui sortent du sommet. La première condition est appelée la condition de continuité, tandis que la seconde est parfois appelée la condition de conservation du courant. En utilisant les notations du théorème précédent, cette condition est écrite grâce aux matrices :

$$A_v = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -1 \\ 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}$$

Et :

$$B_v = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \end{pmatrix}$$

Puisque  $A_v B_v^* = 0$ , on a donc  $A_v B_v^*$  qui est autoadjointe. On vérifie aussi que la matrice  $(A_v B_v)$  a un rang maximal, ainsi, l'hamiltonien est autoadjoint.

## 2.3 EXEMPLES PAR CALCUL DIRECT

On possède des outils pour trouver le spectre de quelques graphes simples. On commencera par un exemple trivial pour comprendre comment appliquer la théorie, puis on augmentera le niveau de complexité. Dans tous les exemples suivants, on utilisera les conditions de Neumann-Kirchhoff.

### 2.3.1 • UN GRAPHE TRIVIAL - UN INTERVALLE

Un intervalle  $[0, L]$  est l'exemple le plus simple d'un graphe ; il a deux sommets (les extrémités de l'intervalle) et une arête. La condition de continuité est vide à chaque sommet car le graphe ne possède qu'une seule arête.



FIGURE 4 – Un graphe trivial

La condition de conservation du courant au sommet 0 donne :

$$f'(0) = 0$$

et au sommet  $L$  donne :

$$-f'(L) = 0$$

En considérant d'abord les valeurs propres positives  $\lambda > 0$ . L'équation des valeurs propres est la suivante :

$$-f'' = k^2 f$$

Pour des raisons de commodité, on prend  $\lambda = k^2$ . Il s'agit d'une équation différentielle linéaire du second ordre avec des coefficients constants, donc la solution s'écrit sous la forme :

$$f(x) = C_1 \cos(kx) + C_2 \sin(kx)$$

En appliquant la condition  $f'(0) = 0$ , on trouve  $C_2 = 0$  et  $f(x) = C_1 \cos(kx)$ . La condition  $f'(L) = 0$  donne donc  $C_1 k \sin(kL) = 0$ , ce qui impose une condition sur  $k$  mais ne permet pas de déterminer  $C_1$  (on n'est naturellement pas intéressés par la solution triviale  $f \equiv 0$ ). On obtient ainsi les valeurs propres  $\lambda = k^2 = (\pi n/L)^2$ ,  $n = 1, 2, \dots$  et les fonctions propres correspondantes  $f(x) = \cos(\pi n x/L)$  définies à un multiplicateur constant près. Si  $\lambda = 0$ , l'équation des valeurs propres devient :

$$f'' = 0$$

Cette équation a pour solution générale  $f(x) = C_1 x + C_2$ . La condition  $f'(0) = 0$  nous donne donc  $C_1 = 0$ , donc  $\lambda = 0$  est une valeur propre et sa fonction propre correspondante est  $f(x) = 1$ , définie à un multiplicateur constant près. Finalement, on traite les valeurs propres négatives,  $\lambda < 0$ . L'équation aux valeurs propres devient :

$$f'' = k^2 f$$

Pour des raisons de commodité, on a prit  $\lambda = -k^2$ . La solution s'écrit ainsi sous la forme :

$$f(x) = C_1 \exp(kx) + C_2 \exp(-kx)$$

La condition  $f'(0) = 0$  donne  $C_1 = C_2$ , et donc  $f(x) = C_1 \exp(kx) + C_1 \exp(-kx)$ . La condition  $f'(L) = 0$  donne  $2kL = 0$  (on ne s'intéresse pas à la solution triviale). On arrive alors à une contradiction, à savoir que les deux conditions  $f'(0) = 0$  et  $f'(L) = 0$  ne sont jamais satisfaites simultanément. L'équation aux valeurs propres n'a donc pas de solution pour  $\lambda < 0$ .

On ne s'est pas intéressé aux valeurs propres complexes. En effet, l'opérateur Hamiltonien défini ci-dessus est autoadjoint et a donc un spectre réel. Le spectre de l'exemple ci-dessus est discret : toutes les valeurs propres sont isolées, et sont de multiplicités finies.

Une façon plus directe d'éliminer les valeurs propres négatives est de noter que l'Hamiltonien utilisée est positif pour le produit scalaire  $\tilde{H}^2(\Gamma)$  et alors on ne peut avoir que des valeurs propres positives.

### 2.3.2 • LE GRAPHE MANDARINE ÉQUILATÉRAL À N ARÊTES

On considère maintenant un graphe à  $n$  arêtes et 2 sommets, comme sur la figure 5, qu'on nomme usuellement graphe mandarine à  $n$  arêtes.

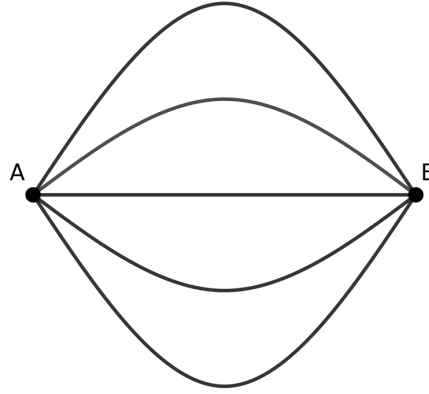


FIGURE 5 – Le graphe mandarin à 5 arêtes et 2 sommets

On numérote les arêtes  $1, \dots, n$ , de haut en bas et on considère le cas où les arêtes ont des longueurs égales à  $L$  (Pour des longueurs générales, les matrices de diffusion - qui seront présentées dans la section suivante - offrent un moyen plus élégant et simple de le faire). Afin de trouver le spectre de ce graphe, on associe des fonctions  $f_1, \dots, f_n$  aux arêtes correspondantes. On a donc  $n$  équations aux valeurs propres  $-f'_i = \lambda f_i$ , pour  $i \in \{1, 2, \dots, n\}$

Puisque l'hamiltonien est autoadjoint positif pour le produit scalaire  $\tilde{H}^2(\Gamma)$  on a que les valeurs propres sont réels et positifs. On pose alors  $\lambda = k^2$  et les équations du valeur propre devient :

$$f'_n = -k^2 f_n$$

Ce qui implique

$$f_i(x) = a_{i1} \cos(kx) + a_{i2} \sin(kx), \text{ pour } i \in \{1, 2, \dots, n\}$$

La condition de continuité en A s'écrit

$$f_i(0) = f_j(0), \text{ pour } i, j \in \{1, 2, \dots, n\}$$

Et alors on a  $a_{i1} = a_{j1} := a$  pour tout  $i, j \in \{1, 2, \dots, n\}$ . La condition de conservation du courant en A donne

$$\sum_{i=1}^n f'_i(0) = 0$$

Donc  $\sum_{i=1}^n a_{i2} = 0$ . La condition de continuité en B s'écrit

$$f_i(L) = f_j(L), \text{ pour } i, j \in \{1, 2, \dots, n\}$$

Et alors  $a_{i2} \sin(kL) = a_{j2} \sin(kL)$ , pour tout  $i, j \in \{1, 2, \dots, n\}$ . En sommant les deux côtés en  $j$  et en utilisant que  $\sum_{i=1}^n a_{i2} = 0$  on trouve que  $a_{i2} \sin(kL) = 0$ , pour tout  $i \in \{1, 2, \dots, n\}$

La condition de conservation du courant en B donne

$$\sum_{i=1}^n f'_i(L) = 0$$

en utilisant encore une fois que  $\sum_{i=1}^n a_{i2} = 0$  on trouve que  $a \sin(kL) = 0$ .

Alors, on a finalement que

$$\begin{cases} a \sin(kL) = 0 \\ a_{i2} \sin(kL) = 0, \text{ pour } i \in \{1, 2, \dots, n\} \end{cases}$$

Supposons que  $\sin(kL) \neq 0$  alors  $a = a_{i2} = 0$  pour tout  $i \in \{1, 2, \dots, n\}$  ce qui ne convient pas. Alors  $\sin(kL) = 0$  et les valeurs propres sont

$$\lambda = k^2 = \left(\frac{\pi n}{L}\right)^2, \quad n = 1, 2, \dots$$

Notons ici un fait remarquable : les fonctions propres dans ce cas ont pour restriction à chaque arête

$$f_i(x) = a \cos\left(\frac{\pi n x}{L}\right) + a_{i2} \sin\left(\frac{\pi n x}{L}\right), \quad \text{avec } \sum_{i=1}^n a_{i2} = 0$$

C'est-à-dire, même si les arêtes sont interchangeables, cela ne veut pas dire que les restrictions d'une fonction propre le sont. Plus explicitement, on peut vérifier que  $f_1(x) = \sin \frac{\pi x}{L}$ ,  $f_2(x) = -\sin \frac{\pi x}{L}$ ,  $f_i \equiv 0$  pour  $i \geq 3$  est bien une fonction propre de ce graphe.

## 2.4 MATRICES DE DIFFUSION ET ÉQUATION SÉCULAIRE

En considérant le graphe comme un système de « fils » homogènes 1D avec des jonctions aux sommets, et en regardant des ondes d'une certaine fréquence se propager le long des fils, on intuite la notion de diffusion au sommet. Du point de vue de la diffusion, les opérateurs du second ordre sont trop restrictifs, tandis que ceux du premier ordre offrent beaucoup plus de flexibilité. Des travaux théoriques montrent qu'on peut résoudre la question portant sur la possibilité de factorisation d'un hamiltonien du second ordre (analogue à la formule  $\Delta = \nabla \cdot \nabla$ ). On peut aussi obtenir une formule de l'indice de Fredholm de tout opérateur différentiel (elliptique) sur un graphe métrique.

### 2.4.1 • GRAPHES QUANTIQUES ET MATRICES DE DIFFUSION

On introduit ici la matrice  $\sigma(v)$  qui décrit la diffusion au sommet  $v$  de l'onde plane  $e^{-ikx}$ , se propageant le long de l'une de ses arêtes adjacentes. Plus précisément, on considère le problème de diffusion pour l'équation suivante :

$$-\frac{d^2 f}{dx^2} = k^2 f(x) \tag{1}$$

Puisque toutes les solutions de l'équation (1) peuvent être représentées comme superposition de  $e^{-ikx}$  et  $e^{ikx}$ , on a :

$$\begin{cases} f^{(e)}(x_e) = e^{-ikx} + \sigma_{e,e}^{(v)} e^{ikx} \\ f^{(e)}(x_{e'}) = \sigma_{e',e}^{(v)} e^{ikx} \text{ en } e' \neq e \\ f \text{ satisfait aux conditions du sommet à } v \end{cases}$$

Il est à noter que les coordonnées  $x$  de chaque arête sont mesurées en partant du sommet  $v$ . On peut interpréter  $\sigma_{e,e}^{(v)}$  comme étant le coefficient de réflexion le long de l'arête  $e$ , tandis que  $\sigma_{e',e}^{(v)}$  avec  $e' \neq e$  est le coefficient de diffusion de l'arête  $e$  à l'arête  $e'$ . Les coefficients  $\sigma_{e,e}^{(v)}$  dépendent en général de  $k$ .

**Définition 2.6.** La matrice  $\sigma^{(v)}(k)$  de taille  $(d_v \times d_v)$  et d'entrées  $\sigma_{e',e}^{(v)}(k)$  est dite être la matrice de diffusion au sommet  $v$ .

**Lemme 2.7.** (1) Si les conditions aux sommets de la première forme du théorème de la section 2.2.2 sont réalisées, on a :

$$\sigma^{(v)}(k) = -(A_v + ikB_v)^{-1} (A_v - ikB_v)$$

(2) Si les conditions aux sommets de la deuxième forme du théorème de la section 2.2.2 sont réalisées, on aura :

$$\begin{aligned} \sigma^{(v)}(k) &= -(U_v - \mathbb{I} + k(U_v + \mathbb{I}))^{-1} (U_v - \mathbb{I} - k(U_v + \mathbb{I})) \\ &= -((1+k)U_v - (1-k))^{-1} ((1-k)U_v - (1+k)) \end{aligned}$$

(3) Si les conditions aux sommets de la troisième forme du théorème 2.5 sont réalisées, on aura :

$$\sigma^{(v)}(k) = -P_{D,v} + P_{N,v} - (\Lambda_v - ik)^{-1} (\Lambda_v + ik) P_{R,v}$$

(4) La matrice  $\sigma^{(v)}(k)$  est unitaire.

**Corolaire 2.8.** (1) La matrice de diffusion  $\sigma^{(v)}$  peut être écrite sous la forme suivante :

$$\begin{aligned} \sigma &= -P_D + P_N - P_R - 2ik(\Lambda - ik\mathbb{I})^{-1} P_R \\ &= -P_D + P_N + P_R - 2(\Lambda - ik\mathbb{I})^{-1} \Lambda P_R \end{aligned}$$

(2) Pour  $k \rightarrow 0$ , on a :

$$\sigma = -P_D + P_N - P_R + O(k)$$

(3) Pour  $k \rightarrow \infty$ , on a :

$$\sigma = -P_D + P_N + P_R + O(k^{-1})$$

**Théorème 2.9.** Pour tout sommet  $v$ , les conditions suivantes sont équivalentes :

- (1)  $\sigma^{(v)}(k)$  est indépendant de  $k$ .
- (2) Il existe une valeur  $k \neq 0$  telle que :  $(\sigma^{(v)}(k))^2 = \mathbb{I}$ .
- (3)  $(\sigma^{(v)}(k))^2 = \mathbb{I}$  pour tout  $k$ .
- (4)  $\sigma^{(v)}$  est sous la forme  $\mathbb{I} - 2Q_v$  pour une certaine projection orthogonale  $Q_v$ .
- (5) Il n'existe aucune partie de Robin à la condition au sommet :  $P_{R,v} = 0$

## 2.4.2 • MATRICE DE DIFFUSION DES LIAISONS ET ÉQUATION SÉCULAIRE

On considère l'équation aux valeurs propres  $\mathcal{H}f = \lambda f$ . On peut poser  $\lambda = k^2$  :

$$-\frac{d^2 f}{dx^2} = k^2 f(x) \tag{2}$$

Soit  $e$  une arête incidente à un sommet  $v$ . Quand  $k \neq 0$ , la solution générale de l'équation (2) peut s'écrire :

$$f(x_e) = C_1 e^{ikx_e} + C_2 e^{-ikx_e} \tag{3}$$

Où  $x_e$  est mesurée à partir du sommet  $v$ . L'arête  $e$  correspond à deux arcs dirigés, l'arc sortant  $b$  et l'arc entrant  $\bar{b}$ . Donc l'équation (2) peut être réécrite comme :

$$f(x_b) = a_b e^{ikx_b} + a_{\bar{b}} e^{ikx_{\bar{b}}} = a_b e^{ikx_b} + e^{ikL_b} a_{\bar{b}} e^{-ikx_b} \quad (4)$$

Où  $x_{\bar{b}}$  est mesurée en partant de l'autre extrémité de l'arête  $e$ , elle est reliée à l'abscisse  $x_b$  par la relation  $x_{\bar{b}} = L_b - x_b$ .

Une fonction propre solution de (2) et vérifiant un certain ensemble de conditions aux limites peut être représentée localement autour d'un sommet  $v$  comme une somme de fonctions  $f_e(x)$ . Les amplitudes sortantes  $a_b$  sont liées aux amplitudes entrantes  $a_{\bar{b}}$  par l'intermédiaire de la matrice de diffusion  $\sigma^{(v)}$ , ce qui est facile à vérifier par un calcul direct. En effet, en notant  $\alpha$  le vecteur  $\{a_b\}$  et  $\beta$  le vecteur  $\{e^{ikL_b} a_{\bar{b}}\}$ , la solution et sa dérivée au sommet  $v$  vérifient  $F(v) = \alpha + \beta$  et  $F'(v) = ik(\alpha - \beta)$ . Les conditions aux limites deviennent alors :

$$A(\alpha + \beta) + ikB(\alpha - \beta) = 0 \quad (5)$$

ce qui est équivalent à :

$$\alpha = -(A + ikB)^{-1}(A - ikB)\beta = \sigma^{(v)}(k)\beta \quad (6)$$

Il est à noter qu'une amplitude sortante d'un sommet en est une entrante à un autre. On combine toutes les conditions aux sommets dans un système de taille  $2E \times 2E$  pour obtenir :

$$\gamma = S(k)e^{ikL}\gamma \quad (7)$$

Où  $\gamma$  est le vecteur constitué des  $2E$  amplitudes entrantes ou sortantes  $a_b$ , et les éléments de la matrice  $S$ , de taille  $(2E \times 2E)$ , sont donnés par :

$$(S)_{b'b} = \delta_{t(b),v} \delta_{o(b'),v} \sigma_{b'b}^{(v)} \quad (8)$$

Avec  $L$  est la matrice diagonale donnée par :

$$L := \begin{pmatrix} L_1 & & & & \\ & \ddots & & & \\ & & L_E & & \\ & & & L_1 & \\ & & & & \ddots \\ & & & & & L_E \end{pmatrix}$$

La matrice  $S(k)$  est appelée matrice de diffusion. L'équation (7) montre que les ondes doivent être en régime permanent pour former une fonction propre de (2). Il est à noter que la dimension de toutes les matrices qu'on vient d'introduire représente le double du nombre total d'arêtes. Le système (2.1.14) possède une solution non nulle si et seulement si :

$$\det(\mathbb{I} - S(k)e^{ikL}) = 0$$

Cette équation est appelée l'équation séculaire du graphe. On résume ces résultats dans le théorème suivant :

**Théorème 2.10.** *On considère le problème aux valeurs propres de l'équation (1), sur un graphe avec des conditions aux sommets données sous l'une des formes du théorème 2.5. Soit  $k \in C \setminus \{0\}$  tel que  $S(k)$  existe. Alors  $\lambda = k^2$  est une valeur propre de  $\mathcal{H}$  si et seulement si  $k$  satisfait l'équation séculaire ci-dessus.*

### 2.4.3 • OPÉRATEURS DU PREMIER ORDRE ET MATRICES DE DIFFUSION

Il est utile de regarder les opérateurs différentiels du premier ordre sur les graphes métriques, en particulier ceux qui sont autoadjoints. L'exemple standard d'un opérateur autoadjoint du premier ordre sur la droite réelle est  $i\frac{\partial}{\partial x}$ . Afin de définir un tel opérateur sur un graphe métrique, on doit supposer que le graphe est orienté. En chaque sommet  $v$ , il existe un certain nombre  $d_v^i$  d'arcs entrants, et un certain nombre  $d_v^o$  qui en sont sortants. On a donc :

$$d_v^i + d_v^o = d_v$$

Étant donné une fonction  $f \in \tilde{H}^1(\Gamma)$ , on considère en chaque sommet  $v$  le vecteur  $F(v)$  des valeurs atteintes en  $v$  par les restrictions de  $f$  aux arêtes adjacentes à  $v$ , qu'on peut décomposer en deux blocs :

$$F(v) = (F^i(v), F^o(v))$$

où  $F^i(v)$  et  $F^o(v)$  sont composés des valeurs atteintes par  $f$  en  $v$  en se restreignant d'abord au arcs entrants, puis au arcs sortants du sommet  $v$ .

On veut imposer des conditions aux sommets qui font de l'opérateur  $D := i \frac{\partial}{\partial x}$  un opérateur auto-adjoint. On considère le cas général où les conditions en chaque sommet  $v$  les conditions sont sous la forme :

$$C_v F(v) = 0$$

où  $C_v$  est un opérateur linéaire dans  $\mathbb{C}^{d_v}$ .

**Théorème 2.11.** *Soit  $\Gamma$  un graphe métrique compact, et  $D := i \frac{\partial}{\partial x}$  défini sur le domaine*

$$\mathcal{D}(D) := \left\{ f \in \tilde{H}^1(\Gamma) \mid C_v F(v) = 0 \text{ pour tous les sommets } v \right\}$$

*On a alors :*

(1) *Les matrices des conditions aux sommets  $C_v$  qui font de l'opérateur  $D$  un auto-adjoint existent si et seulement si pour chaque sommet  $v$  le nombre de les liaisons entrantes et sortantes sont égales, c'est-à-dire  $d_v^i = d_v^o = \bar{d}_v := d_v/2$*

(2) *Si  $d_v^i = d_v^o = \bar{d}_v$  pour chaque sommet  $v$ , les conditions :*

$$F^o(v) = \sigma^{(v)} F^i(v) \tag{9}$$

*où tous les  $\sigma^{(v)}$  sont des matrices unitaires.*

(3) *Inversement, toutes les conditions qui font de  $D$  un opérateur autoadjoint peuvent être écrites sous la forme précédente.*

Tout ensemble de conditions aux sommets linéaires homogènes définit un sous-espace  $L$  sur  $\mathbb{C}^{|\mathcal{B}|}$ . De plus, si les conditions aux sommets sont locales, ce sous-espace est la somme directe des sous-espaces individuels  $L_v \subset \mathbb{C}^{d_v}$  définis pour chaque sommet.

**Lemme 2.12.** *Les conditions aux limites imposées rendent l'opérateur  $D$  symétrique si et seulement si le sous-espace  $L$  qui leur sont associé est un sous-espace isotrope dans  $\mathbb{C}^{|\mathcal{B}|}$  dont la forme limite est une forme symplectique hermitienne. Il est autoadjoint si et seulement si  $L$  est Lagrangien. Dans le cas des conditions locales, ces propriétés sont satisfaites pour les sous-espaces  $L_v$  en chaque sommet  $v$ .*

On introduit le théorème suivant :

**Théorème 2.13.** (1) *Un espace symplectique hermitien  $M$  contient un sous-espace lagrangien si et seulement si sa dimension  $m$  est paire et les indices  $n_-$  et  $n_+$  sont égaux (et donc  $n_- = n_+ = n := \frac{m}{2}$ ).*

(2) *Chaque sous-espace lagrangien a une dimension  $n = m/2$ .*

(3) *La Grassmannienne  $\Lambda_n$  de tous les sous-espaces lagrangiens sur  $M$  peuvent être naturellement identifiés au groupe unitaire  $U(n)$ .*

## 2.5 SPECTRE DE CERTAINS GRAPHES À PARTIR DE L'ÉQUATION SÉCULAIRE

### 2.5.1 • GRAPHE MANDARINE À N ARÊTES

Grâce à l'équation séculaire, on peut trouver le spectre d'un graphe mandarine à  $n$  arêtes d'une manière relativement simple lorsque les arêtes sont de longueurs arbitraires. Soit alors  $L_1, L_2, \dots, L_n$  les longueurs de chaque arête. On note que les conditions de Neumann-Kirchhoff impliquent que la matrice de diffusion  $\sigma$  dans chacun des deux sommets du graphe s'écrit  $\frac{2}{n}\mathbb{E} - \mathbb{I}$ , où  $\mathbb{E}$  est la matrice  $n \times n$  où tous les éléments sont égaux à 1. On pose  $\mathcal{L} = \text{diag}(L_1, L_2, \dots, L_n)$ . Le déterminant de l'équation séculaire s'écrit alors :

$$\begin{aligned} \det \begin{pmatrix} \mathbb{I} & (\mathbb{I} - \frac{2}{n}\mathbb{E})e^{ik\mathcal{L}} \\ (\mathbb{I} - \frac{2}{n}\mathbb{E})e^{ik\mathcal{L}} & \mathbb{I} \end{pmatrix} &= \det \begin{pmatrix} \mathbb{I} & (\mathbb{I} - \frac{2}{n}\mathbb{E})e^{ik\mathcal{L}} \\ 0 & \mathbb{I} - ((\mathbb{I} - \frac{2}{n}\mathbb{E})e^{ik\mathcal{L}})^2 \end{pmatrix} \\ &= \det \left( \mathbb{I} - \left( (\mathbb{I} - \frac{2}{n}\mathbb{E})e^{ik\mathcal{L}} \right)^2 \right) \\ &= \det \left( \mathbb{I} - \left( \mathbb{I} - \frac{2}{n}\mathbb{E} \right) e^{ik\mathcal{L}} \right) \det \left( \mathbb{I} + \left( \mathbb{I} - \frac{2}{n}\mathbb{E} \right) e^{ik\mathcal{L}} \right) \end{aligned}$$

En utilisant l'identité  $\det(A + uv^T) = \det(A) + v^T \text{adj}(A)u$ , où  $\text{adj}(A)$  note la matrice adjointe de  $A$ , on montre que :

$$\begin{aligned} \det \left( \mathbb{I} - \left( \mathbb{I} - \frac{2}{n}\mathbb{E} \right) e^{ik\mathcal{L}} \right) &= \frac{1}{n} \sum_{l=1}^n (1 + e^{ikL_l}) \prod_{\substack{m=1 \\ m \neq l}}^n (1 - e^{ikL_m}) \\ &= \frac{2(-2i)^{n-1}}{n} e^{i\frac{k}{2} \sum_{l=1}^n L_l} \sum_{l=1}^n \cos\left(\frac{kL_l}{2}\right) \prod_{\substack{m=1 \\ m \neq l}}^n \sin\left(\frac{kL_m}{2}\right) \end{aligned}$$

et

$$\begin{aligned} \det \left( \mathbb{I} + \left( \mathbb{I} - \frac{2}{n}\mathbb{E} \right) e^{ik\mathcal{L}} \right) &= \frac{1}{n} \sum_{l=1}^n (1 - e^{ikL_l}) \prod_{\substack{m=1 \\ m \neq l}}^n (1 + e^{ikL_m}) \\ &= \frac{(-2i)2^{n-1}}{n} e^{i\frac{k}{2} \sum_{l=1}^n L_l} \sum_{l=1}^n \sin\left(\frac{kL_l}{2}\right) \prod_{\substack{m=1 \\ m \neq l}}^n \cos\left(\frac{kL_m}{2}\right) \end{aligned}$$

Alors,  $\lambda = k^2$  est valeur propre du graphe si, et seulement si,

$$\sum_{l=1}^n \cos\left(\frac{kL_l}{2}\right) \prod_{\substack{m=1 \\ m \neq l}}^n \sin\left(\frac{kL_m}{2}\right) = 0$$

ou

$$\sum_{l=1}^n \sin\left(\frac{kL_l}{2}\right) \prod_{\substack{m=1 \\ m \neq l}}^n \cos\left(\frac{kL_m}{2}\right) = 0$$

*Remarques.*

— Même si les expressions ci-dessus ne nous donnent pas explicitement le spectre du graphe, on peut en déduire que si le graphe a deux arêtes de même longueur, alors  $\left(\frac{m\pi}{L}\right)^2$  est valeur propre du graphe, où  $L$  est la longueur commune des deux arêtes, pour tout  $m = 1, 2, \dots$ .



- Nous pouvons obtenir une borne inférieure du spectre en notant que, puisque ces expressions ne dépendent que de sinus et de cosinus, elles ne peuvent être nulles (pour  $k > 0$ ) que lorsque  $\frac{kL_i}{2} \geq \frac{\pi}{2}$  et donc les valeurs propres sont bornées inférieurement par  $(\frac{\pi}{l})^2$ , où  $l = \max(L_1, L_2, \dots, L_n)$ . Notez également que nous avons l'égalité dans le cas d'un graphe équilatéral. Cependant, ce n'est pas le seul graphe qui sature la borne. En fait, d'après la remarque précédente, tout graphe qui a deux arêtes de longueurs égales à  $l$  le fait.

### 2.5.2 • GRAPHE ÉTOILE À N ARÊTES

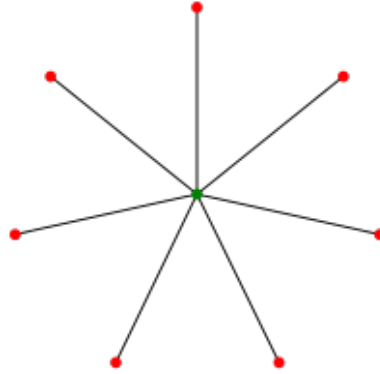


FIGURE 6 – Le graphe étoile à 7 arêtes

On considère cette fois un graphe étoile à  $n$  arêtes (figure 6). Soit  $L_1, L_2, \dots, L_n$  les longueurs de chaque arête. Les conditions de Neumann-Kirchhoff impliquent (par le Lemme 5.1.2) que la matrice de diffusion  $\sigma$  dans le sommet central s'écrit  $\frac{2}{n}\mathbb{E} - \mathbb{I}$ , où  $\mathbb{E}$  est la matrice  $n \times n$  où tous les éléments sont égaux à 1. Les matrices de diffusion aux sommets périphériques sont des matrices  $\mathbb{I} 1 \times 1$ . On pose  $\mathcal{L} = \text{diag}(L_1, L_2, \dots, L_n)$ . Le déterminant de l'équation séculaire s'écrit alors :

$$\begin{aligned} \det \begin{pmatrix} \mathbb{I} & (\mathbb{I} - \frac{2}{n}\mathbb{E})e^{ik\mathcal{L}} \\ -e^{ik\mathcal{L}} & \mathbb{I} \end{pmatrix} &= \det \begin{pmatrix} \mathbb{I} & (\mathbb{I} - \frac{2}{n}\mathbb{E})e^{ik\mathcal{L}} \\ 0 & \mathbb{I} + (\mathbb{I} - \frac{2}{n}\mathbb{E})e^{2ik\mathcal{L}} \end{pmatrix} \\ &= \det \left( \mathbb{I} + \left( \mathbb{I} - \frac{2}{n}\mathbb{E} \right) e^{2ik\mathcal{L}} \right) \\ &= \det \left( \frac{n}{2} (e^{-2ik\mathcal{L}} + \mathbb{I}) - \mathbb{E} \right) \det \left( \frac{2}{n} e^{2ik\mathcal{L}} \right) \end{aligned}$$

En utilisant l'identité  $\det(A + uv^T) = \det(A) + v^T \text{adj}(A)u$ , on montre que :

$$\det \left( \frac{n}{2} (e^{-2ik\mathcal{L}} + \mathbb{I}) - \mathbb{E} \right) = \left( \frac{n}{2} \right)^n \prod_{l=1}^n (e^{-2ikL_l} + 1) - \left( \frac{n}{2} \right)^{n-1} \sum_{l=1}^n \prod_{\substack{m=1 \\ m \neq l}}^n (1 + e^{ikL_m})$$

On a donc :

$$\begin{aligned} \det \begin{pmatrix} \mathbb{I} & (\mathbb{I} - \frac{2}{n}\mathbb{E})e^{ik\mathcal{L}} \\ -e^{ik\mathcal{L}} & \mathbb{I} \end{pmatrix} &= \frac{1}{n} \sum_{l=1}^n (1 - e^{2ikL_l}) \prod_{\substack{m=1 \\ m \neq l}}^n (1 + e^{2ikL_m}) \\ &= \frac{(-2i)^{2^{n-1}}}{n} e^{ik \sum_{l=1}^n L_l} \sum_{l=1}^n \sin(kL_l) \prod_{\substack{m=1 \\ m \neq l}}^n \cos(kL_m) \end{aligned}$$

Alors,  $\lambda = k^2$  est valeur propre du graphe si, et seulement si,

$$\sum_{l=1}^n \sin(kL_l) \prod_{\substack{m=1 \\ m \neq l}}^n \cos(kL_m) = 0$$

Comme dans l'exemple précédent, nous arrivons à une équation implicite pour les valeurs propres du graphe. Bien que l'ensemble de ses solutions pour un cas général ne puisse être obtenu que par des approximations numériques, l'équation obtenue nous permet tout de même de faire quelques observations, qui sont analogues à celles de l'exemple précédent, étant donné la similitude entre les équations trouvées.

*Remarques.*

- On peut montrer que si le graphe a deux arêtes de même longueur, alors  $\left((m + \frac{1}{2}) \frac{\pi}{L}\right)^2$  est valeur propre du graphe, où  $L$  est la longueur commune des deux arêtes, pour tout  $m = 0, 1, \dots$ .
- Nous pouvons obtenir une borne inférieure du spectre en notant que, puisque cette expression ne dépend que de sinus et de cosinus, elle ne peut être nulle (pour  $k > 0$ ) que lorsque  $kL_i \geq \frac{\pi}{2}$  et donc les valeurs propres sont bornées inférieurement par  $(\frac{\pi}{2l})^2$ , où  $l = \max(L_1, \dots, L_n)$ . Notez également que nous avons l'égalité dans le cas d'un graphe équilatéral. Cependant, ce n'est pas le seul graphe qui sature la borne. En fait, d'après la remarque précédente, tout graphe qui a deux arêtes de longueurs égales à  $l$  le fait.

## 2.6 PROPRIÉTÉS SPECTRALES D'UN GRAPHE QUANTIQUE

Le spectre d'un graphe compact peut être détecté par la présence de fonctions propres  $L^2$  (appelées états liés en physique). Quant aux graphes non compacts, ce test ne permet pas de déterminer que ce que l'on appelle le spectre ponctuel  $\sigma_p(\mathcal{H})$ , qui peut être une petite partie (ou même le vide) de l'ensemble du spectre  $\sigma(\mathcal{H})$ . Les autres parties du spectre ne peuvent également pas être déterminées de cette façon. Puisque la notion de graphe quantique  $\Gamma$  inclut un Hamiltonien  $\mathcal{H}$ , nous utiliserons de manière interchangeable les expressions spectre  $\sigma(\mathcal{H})$  de l'opérateur  $\mathcal{H}$  et spectre  $\sigma(\Gamma)$  du graphe quantique  $\Gamma$ .

On présente ci-dessous un ensemble de valeurs spectrales dites particulières, car elles joueront souvent un rôle particulier.

**Définition 2.14.** Nous désignons par  $\sigma_D(\mathcal{H})$  (ou  $\sigma_D(\Gamma)$ ) l'union des spectres de  $\mathcal{H}$  sur les arêtes individuelles avec les conditions de Dirichlet aux sommets. En d'autres termes :

$$\sigma_D(\mathcal{H}) := \{ \lambda \in \mathbb{C} \mid \text{sur une arête } e, \text{ il existe une solution non nulle de l'équation } \mathcal{H}u = \lambda u \text{ qui s'annule sur les extrémités de } e \}$$

Plus explicitement, dans le cas de  $\mathcal{H} = -\frac{d^2}{dx^2}$ , on a :

$$\sigma_D(\mathcal{H}) = \bigcup_{e \in \mathcal{E}} \left\{ \left( \frac{\pi n}{L_e} \right)^2 \mid n \in \mathbb{N} \right\}$$

**Théorème 2.15.** Le spectre  $\sigma(T)$  d'un opérateur auto-adjoint  $T$  dans un espace de Hilbert à dimension infinie est dit de type Sturm-Liouville si  $\sigma(T)$  est constitué de valeurs propres à multiplicités finies qui s'accumulent jusqu'à  $+\infty$ .

Ainsi, on a le théorème suivant :

**Théorème 2.16. (Riesz-Schauder)** Si la résolvante  $(\lambda - T)^{-1}$  d'un opérateur auto-adjoint minoré  $T$  est un opérateur compact, alors le spectre de l'opérateur  $T$  est du type Sturm-Liouville.

**Discrétion du spectre** Nous pouvons donc obtenir le résultat suivant :

**Théorème 2.17.** *Soit  $\Gamma$  un graphe quantique compact équipé de l'Hamiltonien  $\mathcal{H}$  agissant comme  $-d^2/dx^2$  (ou un opérateur de Schrödinger plus général) sur les arêtes et avec des conditions de sommet  $A_v F(v) + B_v F'(v) = 0$ . Ainsi, l'opérateur résolveur  $(\mathcal{H} - i\mathbb{I})^{-1}$  est un opérateur compact en  $L_2(\Gamma)$ . Par conséquent, le spectre  $\sigma(\mathcal{H})$  est du type Sturm-Liouville.*

**Preuve :** En raison du caractère autoadjoint de  $\mathcal{H}$ , le résolveur  $(\mathcal{H} - i\mathbb{I})^{-1}$  existe et est continu de  $L_2(\Gamma)$  vers  $\mathcal{D}(\mathcal{H}) \subset \tilde{H}^2(\Gamma)$ . Le théorème du plongement de Sobolev affirme que le plongement naturel :

$$\tilde{H}^2(\Gamma) \hookrightarrow L_2(\Gamma)$$

est compact. Selon le théorème de Riesz-Schauder, nous pouvons en déduire la conclusion.

**Dépendance des conditions du sommet :** Ici, On déduira quelques bases de la dépendance des valeurs propres du graphe quantique aux paramètres du graphe continu, tels que la matrice  $\mathcal{A} = \{(A_v B_v)\}_{v \in \mathcal{V}}$  des conditions aux sommets, et de la longueur des arêtes  $\{L_e\}_{e \in \mathcal{E}}$ . La notation  $\mathcal{H}_{\mathcal{A}}$  est utilisée pour désigner l'opérateur  $-d^2/dx^2$  sur un graphe compact  $\Gamma$  correspondant à la matrice  $\mathcal{A} = \{(A_v B_v)\}_{v \in \mathcal{V}} \subset U$ . La notation  $\mathcal{H}_{\mathcal{A}, \xi}$  est utilisée pour désigner la version normalisée de  $\mathcal{H}_{\mathcal{A}}$  qui agit comme  $\xi = \xi_e \in (\mathbb{C} \setminus \{0\})^E$  sur le domaine décrit par  $A_\xi = (A_v B_v \Xi_v)$ . Ici  $\xi = \{\xi_e\} \in (\mathbb{C} \setminus \{0\})^E$  est le vecteur constitué des facteurs d'échelle (non-nuls) appliqués sur chaque arête.

Concernant la dépendance du spectre de  $\mathcal{H}_{\mathcal{A}, \xi}$  des paramètres  $(\mathcal{A}, \xi) \subset U \times (\mathbb{C} \setminus \{0\})^E$ , on considère l'opérateur (nommé "pencil" en anglais)  $\mathcal{H}_{\mathcal{A}, \xi} - \lambda \mathbb{I}$  des opérateurs non bornés sur  $L^2(\Gamma)$ . Le résultat ci-dessous porte sur le comportement analytique du spectre.

**Théorème 2.18.** (1) *L'ensemble  $\mathcal{S}$  de tous les vecteurs :*

$$(\mathcal{A}, \xi, \lambda) \subset U \times (\mathbb{C} \setminus \{0\})^E \times \mathbb{C}$$

*tels que que  $\mathcal{H}_{\mathcal{A}, \xi} - \lambda \mathbb{I}$  n'a pas d'inverse borné dans  $L^2(\Gamma)$  est principalement analytique, c'est-à-dire qu'il existe une fonction  $\Phi(\mathcal{A}, \xi, \lambda)$ , analytique sur  $U \times (\mathbb{C} \setminus \{0\})^E \times \mathbb{C}$ , de sorte que  $\mathcal{S}$  coïncide avec l'ensemble de ses zéros.*

(2) *Pour tout entier  $k \geq 1$ , l'ensemble  $\mathcal{S}_k$  de tous les vecteurs :*

$$(\mathcal{A}, \xi, \lambda) \subset U \times (\mathbb{C} \setminus \{0\})^E \times \mathbb{C}$$

*de sorte que  $\dim \text{Ker}(\mathcal{H}_{\mathcal{A}, \xi} - \lambda \mathbb{I}) \geq k$  est analytique.*

Par souci d'exhaustivité, le théorème ci-dessous introduit une conséquence simple et bien connue de la théorie des perturbations.

**Théorème 2.19.** *Le noyau  $\text{Ker}(\mathcal{H}_{\mathcal{A}, \xi} - \lambda \mathbb{I})$  forme un faisceau de vecteurs holomorphes de dimension sur l'ensemble  $\mathcal{S}_k \setminus \mathcal{S}_{k+1}$ . En d'autres termes, si on a une branche analytique locale de valeurs propres  $\lambda(\mathcal{A}, \xi)$  de multiplicité constante  $k$ , alors, au moins localement, on peut choisir une base analytique de  $k$  fonctions propres.*

**Une formule de type Hadamard :** La variation des données spectrales par rapport à la perturbation du domaine est généralement étudiée à l'aide de formules dites variationnelles de Hadamard. Pour simplifier, on considère le cas de la variation des valeurs propres par rapport à une modification de la longueur d'une arête terminale, c'est-à-dire, une arête dont l'une des deux extrémités est une feuille du graphe, la longueur de cette arête sera désignée par la suite par  $s$ .  $s$  désignera également la coordonnée de la feuille située sur l'arête. La preuve suit un modèle bien classique et peut être facilement généralisée.

**Proposition 2.20.** *Soit  $\lambda = \lambda(s)$  une valeur propre simple d'un graphe  $\Gamma$  avec une arête terminale de longueur  $s$  avec la condition de Dirichlet imposée à la feuille  $s$ . Soit  $f = f_s(x)$  la fonction propre normalisée correspondante. On a :*

$$\frac{d\lambda}{ds} = -|f'(s)|^2$$

où  $f'(s)$  est la valeur de la dérivée de la fonction propre à la fin-vertex.

On peut obtenir des résultats similaires en variant les conditions des sommets plutôt que la longueur. Pour des raisons de simplicité, on ne considérera que les conditions aux sommets de type  $\delta$ .

**Proposition 2.21.** Soit  $\lambda = \lambda(s)$  une valeur propre simple d'un graphe  $\Gamma$  qui satisfait la condition de sommet de type  $\delta$  à  $v$  avec le paramètre  $\alpha \neq \infty$ . On a :

$$\frac{d\lambda}{d\alpha} = |f(v)|^2$$

Si on paramètre les conditions à  $v$  comme

$$\zeta \sum_{e \in \mathcal{E}_v} \frac{df}{dx_e}(v) = -f(v)$$

En autorisant maintenant les conditions de Dirichlet ( $\zeta = 0$ ) et en excluant celles de Neumann ( $\zeta = \infty$ ), la dérivé est

$$\frac{d\lambda}{d\zeta} = \left| \sum_{e \in \mathcal{E}_v} \frac{df}{dx_e}(v) \right|^2$$

**La simplicité générique du spectre :** Il existe des résultats bien connus sur la simplicité générique (en ce qui concerne la variation des coefficients de l'opérateur) des spectres des opérateurs elliptiques auto-adjoints sur des variétés compactes. Apparemment, dans "la plupart des situations", le spectre d'un graphe quantique compact est génériquement simple (quand il s'agit de la variation de la longueur des bords). L'expression "la plupart des situations" est introduite en vue d'un simple contre-exemple : un graphe qui est topologiquement un cercle, équipé de conditions Neumann. Dans le cas des conditions aux sommets de type Neumann, il se trouve que c'est la seule exception, comme le montre le résultat suivant de Friedlander, fourni ici sans preuve :

**Théorème 2.22.** Soit  $\Gamma$  un graphe compact avec au moins un sommet de degré différent de 2 et avec l'hamiltonien  $\mathcal{H} = -d^2/dx^2$  équipée de conditions Neumann : les fonctions sont continues aux sommets et :

$$\sum_{e \in \mathcal{E}_v} \frac{df}{dx_e}(v) = 0$$

à n'importe quel sommet  $v$ . Ensuite, pour tout  $\varepsilon > 0$ , on peut modifier les longueurs des arêtes  $L_e$  de  $\Gamma$  par  $\delta L_e < \varepsilon$  pour obtenir un graphe dont toutes les valeurs propres sont simples. De plus, l'ensemble des vecteurs de longueurs  $(L_{e_1}, L_{e_2}, \dots, L_{e_E})$  pour lesquels toutes les valeurs propres de l'Hamiltonien sont simples est un sous-ensemble de la seconde catégorie de Baire dans  $\mathbb{R}_+^E$ .

On supposera que l'Hamiltonien  $\mathcal{H}$  est  $-d^2/dx^2 + V(x)$  avec les conditions aux sommets spécifiées dans les résultats. Les valeurs propres  $\lambda_n$  de  $\mathcal{H}$  (également appelées valeurs propres du graphe et désignées  $\lambda_n(\Gamma)$ ) sont dénombrées dans une suite réelle croissante, en prenant compte de leurs multiplicités. On désigne ici par  $\Gamma_\alpha$  un graphe quantique compact (pas nécessairement connexe) avec un sommet  $v$  qu'on considère parmi l'ensemble de ses sommets. Des conditions arbitraires auto-adjointes sont fixées à tous les sommets autres que  $v$ , tandis que  $v$  est doté de la condition de type  $\delta$  avec un coefficient  $\alpha$  :

$$\begin{cases} f \text{ is continuous at } v \text{ and} \\ \sum_{e \in \mathcal{E}_v} \frac{df}{dx_e}(v) = \alpha f(v) \end{cases}$$

**Théorème 2.23.** Soit  $\Gamma_{\alpha'}$  le graphe obtenu à partir du graphe  $\Gamma_\alpha$  en changeant le coefficient de la condition au sommet  $v$  de  $\alpha$  à  $\alpha'$ . Si  $-\infty < \alpha < \alpha' \leq \infty$  (où  $\alpha' = \infty$  correspond à la condition de Dirichlet), alors :

$$\lambda_n(\Gamma_\alpha) \leq \lambda_n(\Gamma_{\alpha'}) \leq \lambda_{n+1}(\Gamma_\alpha)$$

Si la valeur propre  $\lambda_n(\Gamma_{\alpha'})$  est simple et que sa fonction propre  $f$  est telle que soit  $f(v)$  soit  $f'(v)$  est non nulle, alors les inégalités peuvent être rendues strictes,

$$\lambda_n(\Gamma_\alpha) < \lambda_n(\Gamma_{\alpha'}) < \lambda_{n+1}(\Gamma_\alpha)$$

Ce théorème permet de prouver un critère simple mais utile pour la simplicité des valeurs propres d'un arbre.

**Corolaire 2.24.** Soit  $T$  un arbre avec une condition de type  $\delta$  à chaque sommet interne et une condition de type  $\delta$  étendue à chaque feuille. Si la valeur propre  $\lambda$  de  $T$  admet une fonction propre qui est non nulle sur tous les sommets internes de  $T$ , alors  $\lambda$  est simple. De même, si une valeur propre  $\lambda$  de l'arbre  $T$  est multiple, il existe un sommet interne  $v$  tel que toutes les fonctions de l'espace propre de  $\lambda$  s'annulent sur  $v$ .

Le théorème suivant traite la modification de la structure du graphe en collant une paire de sommets ensemble.

**Théorème 2.25.** *Soit  $\Gamma$  un graphe compact (pas nécessairement connexe). Soit  $v_0$  et  $v_1$  deux sommets du graphe  $\Gamma$  dotés des conditions de type  $\delta$ , c'est-à-dire :*

$$\begin{cases} f \text{ est continue en } v_j \text{ et} \\ \sum_{e \in \mathcal{E}_{v_j}} \frac{df}{dx_e}(v_j) = \alpha_j f(v_j), \quad j = 0, 1 \end{cases}$$

*de telle façon à ce que toutes les conditions aux sommets assurant le caractère auto-adjoint de l'hamiltonien sont autorisées à tous les autres sommets de  $\Gamma$ .*

*Soit  $\Gamma'$  le graphe obtenu à partir de  $\Gamma$  en collant les sommets  $v_0$  et  $v_1$  ensemble en un seul sommet  $v$ , de sorte que  $\mathcal{E}_v = \mathcal{E}_{v_0} \cup \mathcal{E}_{v_1}$ , et doté de la condition de type  $\delta$  :*

$$\sum_{e \in \mathcal{E}_v} \frac{df}{dx_e}(v) = (\alpha_0 + \alpha_1)f(v)$$

*Ainsi, les valeurs propres des deux graphes satisfont les inégalités*

$$\lambda_n(\Gamma) \leq \lambda_n(\Gamma') \leq \lambda_{n+1}(\Gamma)$$

Remarquez que si le domaine de  $h'$  avait pour co-dimension  $k$ , alors on aurait obtenu en appliquant le même argument l'inégalité :

$$\lambda_n(\Gamma) \leq \lambda_n(\Gamma') \leq \lambda_{n+k}(\Gamma)$$

Cette observation conduit immédiatement à la généralisation suivante :

**Théorème 2.26.** *Soit  $\Gamma'$  le graphe obtenu à partir de  $k$  identification de sommets dans  $\Gamma$ , par exemple en collant les sommets  $v_0, v_1, \dots, v_k$  en un seul, ou en collant  $k$  paires en une seule paire de sommets. Chaque identification entraîne également l'ajout des paramètres  $\alpha_j$  dans les conditions de type  $\delta$  du sommet, alors :*

$$\lambda_n(\Gamma) \leq \lambda_n(\Gamma') \leq \lambda_{n+k}(\Gamma)$$

## 2.7 LES FORMULES DE LA TRACE

On explore désormais la connexion entre le spectre d'un graphe quantique et l'ensemble de toutes les orbites périodiques qui le composent. Cette connexion est connue sous le nom de "formule de trace". On suppose que les conditions aux limites n'ont pas de partie de Robin, c'est-à-dire qu'on a  $P_{R,v} = 0$  pour tous les sommets  $v$ . Cela implique en particulier que les valeurs propres du graphe sont positives et que la matrice de diffusion  $S(k)$  est indépendante de  $k$ . Afin de compter les valeurs propres du graphe, on comptera les racines de l'équation séculaire :

$$\det(\mathbb{I} - S(k)e^{ikL}) = 0 \tag{10}$$

où la matrice de diffusion  $S(k)$  est composée des matrices de diffusion  $\sigma^{(v)}$ . On montrera que les racines  $k$  correspondent aux valeurs propres  $\lambda = k^2$ , à une exception près : la valeur propre nulle.

Soit  $K \mapsto N^{sp}(K)$  la fonction de comptage spectral, c'est-à-dire le nombre de valeurs propres  $\lambda = k^2$  du graphe vérifiant  $k < K$ . De plus, soit  $N^{alg}(K_1, K_2)$  le nombre de racines de (10) dans l'intervalle  $(K_1, K_2)$ . Les valeurs propres et les racines sont toutes les deux comptées avec leur multiplicité. Ainsi, les deux fonctions de comptage sont liées par :

$$\#\{\lambda < K^2\} =: N^p(K) = \frac{1}{2}N^{alg}(K, Kxs) - C$$

où  $C$  est une constante qui rend compte de la mauvaise multiplicité de la valeur propre nulle, c'est-à-dire  $C = N_0^{alg}/2 - N_0^{sp}$ .

**Equation séculaire :** Sous l'hypothèse  $S$  ne dépend pas de  $k$ , on peut donner les valeurs propres avec leurs multiplicités correctes :

**Théorème 2.27.** *On considère le problème aux valeurs propres pour l'opérateur  $\mathcal{H}$  agissant comme  $-d^2/dx^2$  sur chaque arête  $e$  avec les conditions aux sommets qui ne comportent pas de partie de Robin ( $P_{R,v} \equiv 0$  sur tous les sommets  $v$ ). On a alors  $k^2 \in \mathbb{C} \setminus \{0\}$  est une valeur propre avec multiplicité  $g$  si et seulement si  $k$  est une racine de (10) avec la même multiplicité.*

On introduit par ailleurs le lemme suivant :

**Lemme 2.28.** *Soit  $S$  une matrice unitaire de taille  $2E \times 2E$ , et soit  $e^{i\theta(k)}$  une valeur propre de la matrice unitaire  $Se^{ikL}$  et  $\vec{u}(k)$  le vecteur propre normalisé correspondant. Alors :*

$$\frac{d\theta}{dk} = \langle \vec{u}(k), L\vec{u}(k) \rangle$$

En particulier,

$$L_{\min} \leq \frac{d\theta}{dk} \leq L_{\max}$$

On définit ici :

$$\zeta(k) = \sqrt{\det(S^*)} \det\left(e^{-i\frac{k}{2}L}\right) \det(\mathbb{I} - Se^{ikL}) \quad (11)$$

Puisque le préfacteur  $\sqrt{\det(S^*)} \det\left(e^{-i\frac{k}{2}L}\right)$  n'est jamais nul, les zéros de  $\zeta(k)$  sont exactement les mêmes que les zéros de l'équation séculaire originale (10). En particulier, ils donnent les valeurs propres du problème spectral correspondant avec les multiplicités correctes (sauf le cas spécial  $k = 0$ ). la fonction  $\zeta(k)$  est analytique et on va maintenant prouver qu'elle ne prend que des valeurs réelles sur la droite réelle.

**Lemme 2.29.** *La fonction analytique  $\zeta(k)$  définie par l'équation (11) est réelle sur la droite réelle. Par conséquent, elle satisfait  $\zeta(\bar{k}) = \overline{\zeta(k)}$  pour  $k \in \mathbb{C}$ .*

**La loi de Weyl :** En utilisant le Lemme précédent, on peut obtenir une estimation du nombre de valeurs propres dans un intervalle donné. Ce type de résultat est souvent appelé "loi de Weyl". On note que les graphes étant essentiellement unidimensionnels, le terme d'erreur est d'ordre 1.

**Lemme 2.30.** *Soit  $(K_1, K_2) \mapsto N(K_1, K_2)$  la fonction de comptage algébrique d'un graphe avec des conditions aux limites ne comportant pas de partie Robin. Alors :*

$$N(K_1, K_2) = \frac{2\mathcal{L}}{2\pi} (K_2 - K_1) + R$$

où  $\mathcal{L} = L_1 + \dots + L_E$  et  $R$  sont uniformément bornés sur  $K_{1,2}$ .

**Dérivation de la formule de la trace :** Désormais, on procède à la dérivation de la formule de la trace en deux étapes : d'abord un développement en terme de traces, ensuite l'évaluation des traces en terme d'orbites périodiques.

**Théorème 2.31.** *Soit  $K \mapsto N(-K, K)$  la fonction de comptage algébrique d'un graphe avec des conditions aux limites ne comportant pas de partie Robin. Alors :*

$$N(-K, K) = 2K \frac{\mathcal{L}}{\pi} + \lim_{\varepsilon \rightarrow 0} \frac{2}{\pi} \operatorname{Im} \sum_{n=1}^{\infty} \frac{1}{n} \operatorname{Tr} \left( Se^{i(K+i\varepsilon)L} \right)^n \quad (12)$$

où la limite est prise ponctuellement pour tous les  $K$  qui ne sont pas les valeurs propres du graphe (et donc les points de discontinuité de la fonction  $N$ ).

On se fixe alors un  $K$  qui n'est pas une valeur propre du graphe. En notant  $\mu_j(\varepsilon)$  la  $j$ -ième valeur propre de la matrice  $Se^{i(K+i\varepsilon)L}$  (qui peut être choisie comme continue dans  $\varepsilon$ ), on remarque que la série dans (12) peut être représentée sous la forme :

$$\sum_{j=1}^{2E} \operatorname{Im} \sum_{n=1}^{\infty} \frac{1}{n} \mu_j^n(\varepsilon)$$

Pour  $\varepsilon = 0$ , les valeurs propres se trouvent sur le cercle des unités, mais ne sont pas égales à 1, car cela signifierait que  $\zeta(K) = 0$ . Ainsi, la série est également convergente pour  $\varepsilon = 0$  bien que pas absolument convergente. On peut prendre la limite terme à terme  $\varepsilon \rightarrow 0$  dans (12) si on arrive à montrer que la série converge uniformément en  $\varepsilon$ . Pour montrer cela, on observe que  $\mu_j(\varepsilon)$  sont bornées loin de 1 et restent dans le cercle fermé de l'unité. Soit  $\delta$  tel que  $|\mu_j(\varepsilon) - 1| \geq \delta$ . On utilise la sommation par parties pour estimer la somme partielle :

$$\begin{aligned} \left| \sum_{n=k}^{k+m} \frac{\mu^n}{n} \right| &= \left| \frac{1}{k+m} \sum_{n=k}^{k+m} \mu^n + \sum_{n=k}^{k+m-1} \left( \frac{1}{n} - \frac{1}{n+1} \right) \sum_{j=k}^n \mu^j \right| \\ &= \left| \frac{1}{k+m} \frac{\mu^k - \mu^{k+m+1}}{1 - \mu} + \sum_{n=k}^{k+m-1} \left( \frac{1}{n} - \frac{1}{n+1} \right) \frac{\mu^k - \mu^{n+1}}{1 - \mu} \right| \\ &\leq \frac{1}{k+m} \frac{2}{\delta} + \sum_{n=k}^{k+m-1} \left( \frac{1}{n} - \frac{1}{n+1} \right) \frac{2}{\delta} = \frac{2}{k\delta} \end{aligned}$$

la convergence uniforme en découle immédiatement.

### 3

## APPROCHE NUMÉRIQUE DU PROBLÈME

### 3.1 MÉTHODE VARIATIONNELLE ET ÉLÉMENTS FINIS

Soit  $\Gamma = (V, E)$  un graphe métrique sur lequel on considère l'opérateur  $\mathcal{H} : u(x) \rightarrow -\frac{d^2 u}{dx^2}$ . On peut poser alors :

$$h : H_1(\Gamma) \times H_1(\Gamma) \rightarrow \mathbb{R},$$

$$h(u, g) = \sum_{e \in E} \int_e \frac{du}{dx} \frac{dg}{dx} dx$$

qui est la forme bilinéaire qui correspond à l'opérateur  $\mathcal{H}$  sous les conditions de Neumann-Kirchhoff. C'est-à-dire :

$$\sum_{e \in E_v} \frac{du}{dx}(v) = 0, \forall v \in V$$

avec :  $E_v$  le sous-ensemble de  $E$  qui contient les arêtes incidentes au sommet  $v$ . La fonction énergie qui correspond à cette forme est :

$$J(u) = \frac{1}{2} \sum_{e \in E} \int_e \left| \frac{du}{dx} \right|^2 dx, \forall u \in H^1(\Gamma)$$

Pour  $f \in L^2(\Gamma)$ , on cherche une solution à l'équation :

$$h(u, g) = \int_{\Gamma} f(x)g(x)dx, \forall g \in H^1(\Gamma)$$

À l'aide de l'inégalité Poincaré, on montre que la forme bilinéaire symétrique est coercive ; elle est bien continue, l'équation a donc une et une seule solution selon le théorème de Lax-Milgram. Cette unique solution minimise l'énergie  $J$ . Sur chaque arête du graphe, il est possible d'utiliser la méthode des éléments finis unidimensionnel afin d'approcher la solution.

Soit  $e$  une arête définie par ses deux extrémités, qu'on note  $V_a^e$  et  $V_b^e$ . La première étape consiste à subdiviser l'arête  $e$  en  $n_e$  segments de longueurs  $h_e$ , avec  $n_e \geq 2$ . Les points

$$V_e = \{x_j^e\}_{j=0}^{n_e-1} \cup \{V_a^e\} \cup \{V_b^e\}$$

forment une chaîne contenue dans l'arête  $e$ , reliant le sommet  $V_a^e$  au sommet  $V_b^e$ . Les points intérieurs  $x_j^e$  sont appelés les noeuds de la discrétisation. On désigne par  $\{\Psi_j^e\}_{j=1}^{n_e-1}$  la base du chapeau des fonctions standard, c'est à dire :

$$\Psi_j^e(x^e) = \begin{cases} 1 - \frac{|x_j^e - x^e|}{h_e}, & \text{si } x_{j-1}^e \leq x^e \leq x_{j+1}^e \\ 0, & \text{sinon} \end{cases}$$

où on a fixé  $x_0^e = V_a^e$  et  $x_{n_e}^e = V_b^e$ . On définit également le voisinage  $W_v$  d'un sommet  $v \in V$  comme l'ensemble union de tous les ensembles de la forme  $[v_a^e, x_1^e]$  ou  $[x_{n_e}^e, v_b^e]$  :

$$W_v = \left( \bigcup_{e \in \{e \in E_v \text{ tq } v_a^e = v\}} [v, x_1^e] \right) \cup \left( \bigcup_{e \in \{e \in E_v \text{ tq } v_b^e = v\}} [x_{n_e-1}^e, v] \right)$$

$W_v$  est lui-même un graphe métrique (en forme d'étoile), et en tant que sous-graphe, il hérite des propriétés métrique du graphe original. On introduit les fonctions  $\Phi_v(x)$ , de supports  $W_v$ , définies par :

$$\Phi_v(x)|_{W_v \cap e} = \begin{cases} 1 - \frac{|x_v^e - x^e|}{h_e}, & \text{si } x^e \in W_v \cap e; e \in E_v \\ 0, & \text{sinon} \end{cases}$$

où  $x_v^e$  est 0 ou  $l_e$  en fonction de la direction de l'arête et de son paramétrage. Les fonctions  $\Psi_j^e$  sont une base de l'espace de dimension finie  $V_h^e$  défini par :

$$V_h^e = \left\{ w \in H_0^1(e); w|_{[x_j^e, x_{j+1}^e]} \in P_1, j = 0, 1, \dots, n_e - 1 \right\}$$

où  $P_1$  est l'espace des fonctions linéaires. Globalement, nous construisons l'espace :

$$V_h(\Gamma) = \left( \bigoplus_{e \in E} V_h^e \right) \oplus Vect\{\Phi_v\}_{v \in V}$$

et on a :

$$V_h(\Gamma) \subset C^0(\Gamma)$$

Il s'agit d'un sous-espace de  $H^1(\Gamma)$  de dimension fini. Ainsi, toute fonction  $W_h \in V_h(\Gamma)$  est une combinaison linéaire des fonctions  $\Psi_j^e$  et  $\Phi_v$  :

$$W_h(x) = \sum_{e \in E} \sum_{j=1}^{n_e-1} \alpha_j^e \Psi_j^e(x) + \sum_{v \in V} \beta_v \Phi_v$$

Lorsqu'on approche l'équation variationnelle :

$$h(u, g) = \int_{\Gamma} f g dx, \forall g \in H^1(\Gamma)$$

avec  $f \in L^2(\Gamma)$ , sur  $V_h$  on test l'équation ci-dessus en remplaçant  $g$  par chaque  $\Psi_j^e, \Phi_v$ . Mais puisque  $W_v \cap W_z = \emptyset$  dès que  $v \neq z$ , on obtient le système d'équations suivantes :



$$(*) \begin{cases} h_h(W_h, \Psi_k^e) &= \sum_{e \in E} \sum_{j=1}^{n_e-1} \alpha_j^e \int_e \frac{d\Psi_j^e}{dx} \frac{d\Psi_k^e}{dx} dx + \sum_{v \in V} \beta_v \int_{W_v} \frac{d\Psi_j^e}{dx} \frac{d\Phi_v}{dx} dx \\ &= \int_{\text{supp}(\Psi_j^e)} f \Psi_j^e dx, \forall \Psi_j^e; \\ h_h(W_h, \Phi_v) &= \sum_{e \in E} \sum_{j=1}^{n_e-1} \alpha_j^e \int_{W_v} \frac{d\Psi_j^e}{dx} \frac{d\Phi_v}{dx} dx + \sum_{v \in V} \beta_v \int_{W_v} \frac{d\Phi_v}{dx} \frac{d\Phi_v}{dx} dx \\ &= \int_{W_v} f \Phi_v dx, \forall \Phi_v \end{cases}$$

Sous les hypothèses de la section précédente (continuité et coercivité du Hamiltonien), le système (\*) admet une solution, qui est en plus unique, sur  $V^h(\Gamma)$ , et qu'on notera dans la suite  $u_h \in V_h(\Gamma)$ . Le théorème suivant donne quelques propriétés du système d'équations (\*) :

**Théorème 3.1.** *On pose :*

$$u_E = \begin{bmatrix} u^1 \\ \vdots \\ u^M \end{bmatrix}, \text{ où } u^e = \begin{bmatrix} u_1^e \\ \vdots \\ u_{n_e-1}^e \end{bmatrix}$$

Et :

$$u_V = \begin{bmatrix} u^1 \\ \vdots \\ u^N \end{bmatrix}$$

Et on considère la fonction de  $V_h(\Gamma)$  définie par :

$$u_h(x) = \sum_{e \in E} \sum_{j=1}^{n_e-1} u_j^e \Psi_j^e + \sum_{v \in V} u_v \Phi_v(x), x \in \Gamma$$

De la même manière, on considère les vecteurs  $\mathbf{f}_E$  et  $\mathbf{f}_V$  définis par :

$$\mathbf{f}_E = \begin{bmatrix} \mathbf{f}^1 \\ \vdots \\ \mathbf{f}^M \end{bmatrix}, \text{ où } \mathbf{f}^e = \begin{bmatrix} f_1^e \\ \vdots \\ f_{n_e-1}^e \end{bmatrix}$$

$$\mathbf{f}_V = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix}$$

Avec  $f_k^e = \int_{\text{supp}(\Psi_k^e)} f \Psi_k^e dx$  et  $f_v = \int_{W_v} f \Phi_v dx$ . Le système (\*) peut ainsi être écrit sous la forme :

$$\mathbf{H} \begin{bmatrix} \mathbf{u}_E \\ \mathbf{u}_V \end{bmatrix} = \begin{bmatrix} \mathbf{f}_E \\ \mathbf{f}_V \end{bmatrix}$$

Où  $\mathbf{H}$  se décompose sous la forme suivante :

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{12}^T & \mathbf{H}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{G} \end{bmatrix} = \mathbf{L}$$

De plus :

1.  $\mathbf{H}$  est une matrice symétrique définie positive.
2.  $\mathbf{A}$  est une matrice diagonale par blocs, et chaque bloc (de dimension  $n_e - 1$ ) peut être changé en une matrice symétrique tridiagonale non singulière, qui correspond à l'arête  $e$ .

3. Les éléments de  $\mathbf{A}$  sont donnés par :

$$\int_e \frac{d\Psi_j^e}{dx} \frac{d\Psi_k^e}{dx} dx = \begin{cases} 2/h_e, & \text{si } j = k \\ -1/h_e, & \text{si } |j - k| = 1 \\ 0, & \text{sinon} \end{cases}$$

Tandis que les éléments de  $\mathbf{B}$  sont données par :

$$\int_{W_v} \frac{d\Phi_v}{dx} \frac{d\Psi_k^e}{dx} dx = \begin{cases} -1/h_e, & \text{si } e \in W_v \cap E_v \neq \emptyset \\ 0, & \text{sinon} \end{cases}$$

De plus,  $\mathbf{G}$  est diagonale, et ses éléments sont donnés par :

$$\int_{W_v} \frac{d\Phi_v}{dx} \frac{d\Phi_v}{dx} dx = \sum_{e \in E_v} 1/h_e$$

**Théorème 3.2.** Soit  $f \in L^2(\Gamma)$ , alors la solution approximative  $u_h$  de (\*) vérifie l'inégalité :

$$\|u - u_h\|_{H^1(\Gamma)} \leq \sqrt{\frac{\gamma_1}{\gamma_0}} \Theta C \hat{h} \sum_{e \in E} \|u|_e\|_{H^2(e)}$$

Où  $\hat{h} = \max_e h_e$ ,  $C$  est une constante indépendante de  $u$  et  $\hat{h}$  et  $\Theta = \text{Vol}(\Gamma) = \sum_{e \in E} l_e$ , le volume de  $\Gamma$ .

**Preuve :** On a :

$$h(z, w) = \int_{\Gamma} \frac{dz}{dx} \frac{dw}{dx} dx$$

est coercive et continue :

$$\begin{cases} \gamma_0 \|z\|_{H^1(\Gamma)}^2 \leq h(z, z) \\ |h(z, w)| \leq \gamma_1 \|z\|_{H^1(\Gamma)} \|w\|_{H^1(\Gamma)} \end{cases}$$

À partir de ces inégalités, on voit que la norme énergie  $\|z\|^2 = h(z, z)$  et la norme de  $H^1(\Gamma)$  sont équivalentes. Et du fait que  $V_h(\Gamma) \subset H^1(\Gamma)$ , on alors  $z = u - u_h$  minimise la norme énergie de  $z$  en  $V_h(\Gamma)$  et on :

$$\gamma_0 \|u - u_h\|_{H^1(\Gamma)}^2 \leq h(z, z) \leq h(u - u_h^I, u - u_h^I) \leq \gamma_1 \|u - u_h^I\|_{H^1(\Gamma)}^2$$

Où  $u_I$  est l'interpolation de  $u$  en les noeuds et les sommeils. On observe aussi sous ces conditions que la solution de (\*) satisfait  $u|_e \in H^2(e)$  en chaque arête  $e \in E$ . En chaque arête  $e = (v_a, v_b)$ , les  $\psi_j^e$ , ( $j = 1, \dots, n_e - 1$ ) et la restriction sur  $e$  de  $\phi_{v_a}$  et  $\phi_{v_b}$  forment une base de l'approximation par les éléments finis de  $H_1(e)$ . Donc, on a l'estimation de l'erreur type

$$\|u|_e - u_h|_e\|_{H^1(e)} \leq \sqrt{\frac{\gamma_1}{\gamma_0}} l_e C h_e \|u|_e\|_{H^2(e)} \quad \forall e \in E$$

avec  $C$  une constante indépendante de  $u$  et de  $h_e$ . En prenant maintenant  $\hat{h} = \max_e h_e$ , et en sommant sur toutes les erreurs locaux et leurs limites supérieurs on arrive à l'inégalité voulue pour  $\|u - u_h\|_{H^2(\Gamma)}$ .

## 3.2 MÉTHODE DE LA RÉOLUTION DE L'ÉQUATION SÉCULAIRE

Dans cette partie, et afin de pouvoir utiliser l'outil numérique pour le calcul des valeurs propres de l'opérateur laplacien positif sur un graphe métrique donné, il nous faut d'abord pouvoir représenter ce dernier. Pour cela, nous avons opté pour une solution moins conventionnelle que celles se basant sur la matrice d'adjacence, à savoir la représentation du graphe par un objet d'une classe, classe qu'on aura donc à définir. Pour ce qui est de la structure métrique, la longueur de chaque arête doit être portée par au moins un attribut de cette classe, afin de pouvoir représenter correctement un graphe quantique (se référer à la section 5.1 de l'appendice).

### 3.2.1 • DESCRIPTION DE LA MÉTHODE

Comme il est montré dans la section 2.4.2, les valeurs propres (notées  $k^2$ ) de l'opérateur laplacien positif d'un graphe métrique connexe  $(G, V, L)$ , sont solution de l'équation suivante :

$$(i) \det (\mathbb{I} - S(k)e^{ikL}) = 0$$

Où :

- $\mathbb{I}$  est la matrice identité de taille  $2E$ , avec  $E = |V|$  le nombre d'arêtes du graphe.
- $L$  est une matrice diagonale de taille  $2E$ , où le  $i$ ème élément diagonal est la longueur du la  $i$ ème arête du graphe orienté issu de  $G$  après dédoublement de ses arêtes.
- $S$  est la matrice carrée de taille  $2E$  définie par :

$$S_{(b,b')}(k) = \delta_{(t(b),v)} \delta_{(o(b'),v)} \sigma_{(b',b)}^{(v)}(k)$$

Avec :

$$\sigma^{(v)}(k) = -(A_v + ikB_v)^{-1} (A_v - ikB_v)$$

L'équation (i) étant l'équation séculaire du graphe.

On voit alors qu'en construisant la matrice  $S$ , puis en résolvant l'équation séculaire, on a accès aux valeurs propres de l'opérateur laplacien positif. C'est sur ce principe qu'on s'est basé afin d'implémenter la méthode `.k1()` de la classe `Graph` qui permet d'obtenir la plus petite valeur propre non nulle d'un graphe donné (se référer à la section 5.1 de l'appendice). Sans rentrer dans les détails du code, cette méthode fait, dans l'ordre, les opérations suivantes :

- Construire les matrices  $A^v$  et  $B^v$  qui spécifient les conditions aux limites locales, cela pour chaque sommet du graphe (les deux méthodes `.A()` et `.B()` de la classe `Graph`).
- Calculer les matrices  $\sigma^v(k)$ , cela pour chaque sommet du graphe (la méthode `.sigma()` de la classe `Graph`).
- Calculer l'ensemble des arêtes orientées du graphe considéré (la méthode `.bonds()` de la classe `Graph`).
- Construire la matrice de diffusion  $S(k)$  (la méthode `.S()` de la classe `Graph`).
- Calculer le membre gauche de l'équation séculaire sous la forme (i) (la méthode `.seqEq()` de la classe `Graph`).
- Résoudre l'équation séculaire par minimisation de la valeur absolue du membre gauche de l'équation séculaire sous la forme (i) (méthode `.k1()` de la classe `Graph`).

Il faut noter toutefois que le code se restreint par défaut à des conditions de type Neumann pour réaliser l'étape 1. Cette restriction permet de simplifier relativement les calculs puisqu'on peut alors se restreindre, sans perte de généralité, aux graphes dont la longueur totale de ses arêtes vaut 1 (Les conditions aux limites de type Newman sont invariantes par changement d'échelle).

Il en suit alors une méthodes de calcul des valeurs propres du laplacien positif, se basant sur la résolution de l'équation séculaire. Il faut donc tester le code sur des graphes pour lesquels on peut calculer la première valeur propre, ou alors comparer les valeurs obtenues pour des graphes bien spécifiques avec celles obtenues par d'autres méthodes numériques indépendamment implémentées (la méthode variationnelle ou probabiliste).

### 3.2.2 • TESTS DU CODE SUR DES GRAPHEs "SIMPLES"

Le test le plus trivial, et que tout bon code doit pouvoir passer, est le calcul de la première valeur propre d'un graphe segment unitaire (figure 4). Formellement, ce graphe peut être construit de la façon suivante (figure 7) :

```
>>> G = Graph()

>>> G.addNewVertice()
0

>>> G.addNewVertice()
1

>>> G.addEdge(0,1)

>>> G.k1()
3.1415868667530473
```

FIGURE 7 – Construction et test numérique sur le graphe segment unitaire.

Pour chercher les valeurs propres du laplacien positif de ce graphe, une façon rapide est de noter  $f$  une fonction propre quelconque de ce dernier pour une valeur propre qu'on notera  $k^2$ , dès alors on peut écrire que :  $f(x_b) = \alpha e^{(ikx_b)} + \beta e^{(ikx_{\bar{b}})}$ , où :  $x_{\bar{b}} = 1 - x_b$ , le fait que le problème de recherche des valeurs propres soit un problème linéaire nous permet de pouvoir choisir  $\alpha = 1$ , soit choisir  $f$  sous la forme :  $f(x_b) = e^{(ikx_b)} + \beta e^{(ikx_{\bar{b}})}$ . On peut remarquer que le graphe est invariant sous l'action de l'opérateur qui échange les sommet 0 et 1, cela impose le fait que cet opérateur commute avec l'opérateur laplacien positif du graphe (son hamiltonien). Un tel opérateur est idempotent, ses valeurs propres sont donc soit -1 soit 1, on peut ainsi se restreindre dans la recherche des fonctions propres du laplacien positif à celle qui vérifient ou bien  $f(x_b) = f(x_{\bar{b}})$ , ou bien  $f(x_b) = -f(x_{\bar{b}})$ , et donc ou bien  $\beta = 1$  ou bien  $\beta = -1$ . Ainsi, on peut faire cette recherche pour  $f_+(x_b) = e^{(ikx_b)} + e^{(ikx_{\bar{b}})}$  et  $f_-(x_b) = e^{(ikx_b)} - e^{(ikx_{\bar{b}})}$ . Les conditions aux limites de type Newman sur ce graphe imposent que :  $f'(0) = f'(1) = 0$ , ce qui impose pour  $f_+$  qu'on ait :  $1 - e^{ik} = 0$ , soit :  $k = 2\pi n, n \in \mathbb{N}$ , et pour  $f_-$  qu'on ait :  $1 + e^{ik} = 0$ , soit  $k = \pi + 2\pi n, n \in \mathbb{N}$ , finalement, on voit alors que le spectre du laplacien positif pour ce graphe est  $\{k^2/k \in \pi\mathbb{N}\}$ , sa plus petite valeur non nulle correspond donc à  $k = \pi$ .

Lorsqu'on exécute le code qui résout l'équation séculaire sur ce graphe, on obtient  $\pi$  à quelques décimales près (figure 7).

D'autres graphes pour lesquels on sait établir des résultats exacts sont les graphes de type « mandarine » (figure 5). Pour ce type de graphes, il est clair qu'en raison de symétrie de rôle des arêtes, la répartition de longueurs qui maximise la première valeur propre est l'équi-répartition. C'est pour la même raison que la fonction propre de l'hamiltonien qui correspond au maximum de la première valeur propre admet pour restriction la même fonction sur toutes les arêtes du graphes. Si on note  $f$  un élément qui engendre un tel sous-espace propre, alors comme évoqué plus haut, on peut prendre  $f$  sous la forme :  $f(x_b) = e^{(ikx_b)} + \beta e^{(ikx_{\bar{b}})}$  avec  $\beta$  qui ne dépend pas de l'arête  $b$ , et où on note cette fois  $x_{\bar{b}} = 1/|E| - x_b$ . Encore une fois, par le même argument d'invariance du graphe sous échange du sommet 0 et du sommet 1, on se restreindre dans notre recherche encore à  $f_+$  et  $f_-$ . On peut encore une fois montrer que les conditions aux limites de type Newman pour ce graphe sont équivalentes à  $f'(0) = f'(1) = 0$ , chose qui impose pour  $f_+$  qu'on ait :  $1 - e^{(ik/|E|)} = 0$ , soit :  $k = 2\pi|E|n, n \in \mathbb{N}$ , et pour  $f_-$  qu'on ait :  $1 + e^{(ik/|E|)} = 0$ , soit  $k = \pi|E| + 2\pi|E|n, n \in \mathbb{N}$ , finalement, la valeur minimale de la plus petite valeur propre non nul du laplacien positif sur un graphe « mandarine » est  $(|E|\pi)^2$ .

Peut-on retrouver cette valeur grâce à une simulation numérique ? Oui mais pas en représentant le graphe à l'aide d'une matrice d'adjacence. En effet, les graphes de type « mandarine » ont la particularité d'être des graphes qui possèdent des arêtes multiples. Pour représenter un graphe qui contient des boucles sur des sommets ou des arêtes multiples entre deux sommets, il faut se permettre des entiers quelconques sur la matrice d'adjacence, et non seulement des 0 et des 1. C'est pour éviter ce genre de complications qu'on procède de la façon suivante :

### 3.2.3 • TRAITEMENT DU CAS DES GRAPHS À BOUCLES OU À ARÊTES MULTIPLES

Souvent, dans la théorie des graphes, on distingue un type de sommets bien particuliers qu'on qualifie de sommets « simples ». Dans le cas des graphes non orientés, ce sont les sommets dont l'adjacence vaut exactement 2 (figure 8). Pour ce qui est des graphes orientés, ce sont les sommets dont l'adjacence entrante et l'adjacence sortante valent exactement 1 (figure 9).

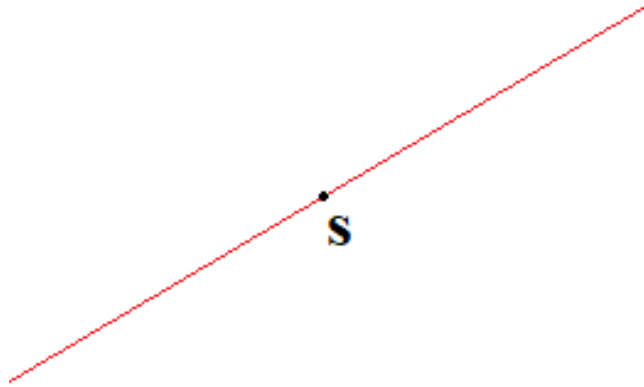


FIGURE 8 – Sommet simple dans un graphe non orienté..

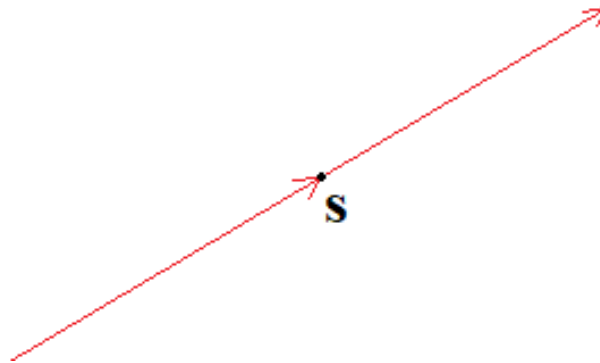


FIGURE 9 – Sommet simple dans un graphe orienté.

On qualifiera dans la suite de graphe dérivé d'un graphe métrique donné, tout graphe qui est soit le graphe original, ou un graphe qui peut être obtenu à partir d'un graphe dérivé du graphe original en introduisant un nouveau sommet simple en une abscisse  $x_b$  sur l'arête orientée  $b$ , et en créant ainsi deux nouvelles arêtes orientées, dont la première a pour origine l'origine de  $b$  et pour terminaison le nouveau sommet et de longueur  $x_b$ , et dont la deuxième a pour origine le nouveau sommet et pour terminaison la terminaison de  $b$  et de longueur  $x_{\bar{b}}$ . On étendra ce concept aux graphes non orientés en recollant les arêtes orientées opposées après orientation des arêtes et dérivation. La figure 10 est un exemple de graphe dérivé du graphe de la figure 5.

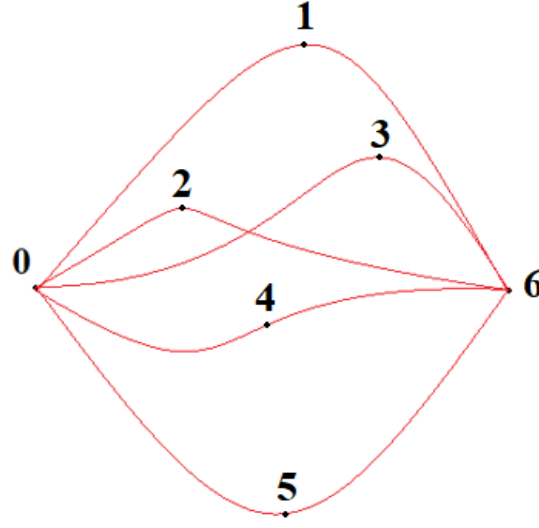


FIGURE 10 – Exemple de graphe dérivé du graphe de la figure 7.

Ce concept est intéressant du fait du résultat suivant :

**Lemme 3.3.** *Le spectre d'un graphe quantique est invariant sous dérivation.*

Bien que les graphes dérivés ont plus de sommets que le graphe original, ce qui fait que leurs matrices d'adjacences sont nécessairement plus grandes que celle du graphe original, toutefois, si on introduit suffisamment de sommets simples sur les arêtes qui constituent une boucle sur un sommet ou qui sont multiples entre deux sommets, on peut alors éliminer la boucle ou la multiplicité des arêtes, et donc, après suffisamment de dérivations, on peut se retrouver avec un graphe sans boucles ni arêtes multiples. En effet, sur le graphe de la figure 10, on peut remarquer l'absence des arêtes multiples contrairement au graphe de la figure 5.

La méthode `.resolve()` de la classe `Graph()` (voir l'appendice) procède à une suite de dérivations sur le graphe original, chose qui permet d'éliminer les arêtes multiples et les boucles aux sommets, et ainsi de procéder au calcul de la matrice de diffusion sans ambiguïté.

Une simple simulation numérique permet de retrouver la valeur prédite théoriquement pour la première valeur propre non nulle d'un graphe mandarine à 5 arêtes par exemple, censée être  $5\pi$  et confirmée par l'outil numérique (figure 11) :

```
>>> G = Graph()
G.addNewVertice()
G.addNewVertice()
G.addEdge(0,1)
G.addEdge(0,1)
G.addEdge(0,1)
G.addEdge(0,1)
G.addEdge(0,1)
G.normalize()

print("Valeur propre initiale trouvee :"+str(G.k1()/pi)+"pi")
0
1
Valeur propre initiale trouvee :4.999998218046216pi
```

FIGURE 11 – Construction et test numerique sur le graphe mandarine à 5 arêtes.

### 3.2.4 • UTILISATION DE L'OUTIL NUMÉRIQUE POUR DES PROBLÈMES OUVERTS

L'un des problèmes encore ouverts de la théorie des graphes quantiques est la valeur maximale (par rapport à la structure métrique du graphe) de la première valeur propre de l'opérateur laplacien positif sur un graphe mandarine, mais qu'on a modifié en rajoutant une boucle :

Sur l'un des deux sommets (figure 12) :

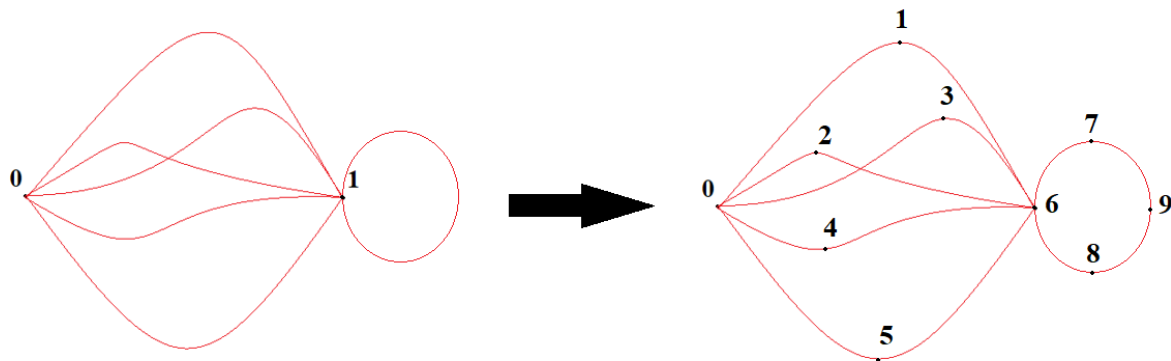


FIGURE 12 – Rajout d'une boucle sur l'un des deux sommets d'un graphe "mandarine" à cinq arêtes, et son graphe dérivé qu'on utilisera.

Sur les deux sommets (figure 13) :

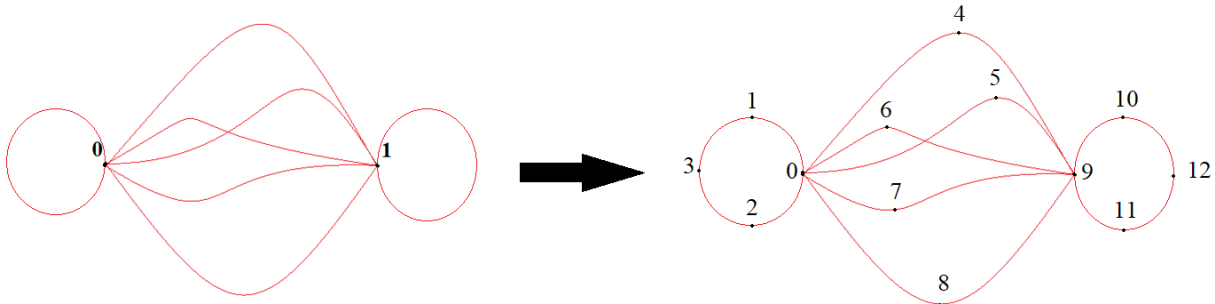


FIGURE 13 – Rajout de deux boucles sur les deux sommets d’un graphe ”mandarine” à cinq arêtes, et son graphe dérivé qu’on utilisera.

On essaiera de traiter numériquement ces deux cas :

**Premier cas : rajout d’une seule boucle (figure 14).**

En contractant la boucle qu’on a rajouté au graphe mandarine à 5 arêtes, on peut voir que le maximum de la première valeur propre du graphe considéré est au moins  $(5\pi)^2$ . Un théorème de [6] nous permet d’affirmer que ce maximum ne peut dépasser  $(6\pi)^2$ .

```
>>> (executing file "main.py")
Valeur propre initiale trouvée :4.142865690174999pi
...
Optimization terminated successfully.
    Current function value: -17.419526
    Iterations: 390
    Function evaluations: 803
Valeur propre optimale trouvée :5.544807303017045pi
```

FIGURE 14 – Résultat de la simulation sur un graphe mandarine à 5 arêtes auquel on a rajouté une seule boucle.

Grâce à la simulation de la figure 14, on peut voir que la petite valeur propre de ce graphe part d’une valeur voisine de  $(4.14\pi)^2$  (pour le graphe équilatéral) vers une valeur voisine de  $(5.54\pi)^2$ , cela après 390 itérations. D’une part, on est donc sûr que le maximum de la plus petite valeur propre de ce graphe est strictement supérieur à  $(5\pi)^2$ , et d’autre part, on obtient une valeur propre qui correspond à la géométrie décrite par la figure 15 :



```
>>> (executing file "main.py")
Géométrie initiale du graphe :
{0: [(1, 0.181818181818182), (1, 0.181818181818182), (1, 0.181818181818182), (1, 0.181818181818182), (1, 0.181818181818182)], 1: [(0, 0.181818181818182), (0, 0.181818181818182), (0, 0.181818181818182), (0, 0.181818181818182), (0, 0.181818181818182), (0, 0.181818181818182), (1, 0.181818181818182)]}
Valeur propre initiale trouvée : 4.142865690174999pi
...
Optimization terminated successfully.
Current function value: -17.419521
Iterations: 361
Function evaluations: 763
Valeur propre optimale trouvée : 5.544805726394676pi
Géométrie du graphe optimal :
{0: [(1, 0.180606480247977), (1, 0.18257756964027352), (1, 0.18041018389634517), (1, 0.18199719520109187), (1, 0.18030198895048635)], 1: [(0, 0.180606480247977), (0, 0.18257756964027352), (0, 0.18041018389634517), (0, 0.18199719520109187), (0, 0.18030198895048635), (1, 0.1882131641276527)]}
```

FIGURE 15 – Résultat de la simulation sur un graphe mandarine à 5 arêtes auquel on a rajouté une seule boucle : les géométries des graphes initial et optimal sont décrites par un dictionnaire de longueurs.

Il peut paraître curieux que la géométrie du graphe optimal s'écarte très peu de la géométrie du graphe initial alors que les premières valeurs propres s'écarte considérablement. Cela nous pousse à vérifier que l'algorithme ne capture pas une valeur propre d'ordre plus élevé. C'est pourquoi on trace la valeur absolue du membre gauche de l'équation séculaire en fonction de la variable  $k$  (figure 16) :

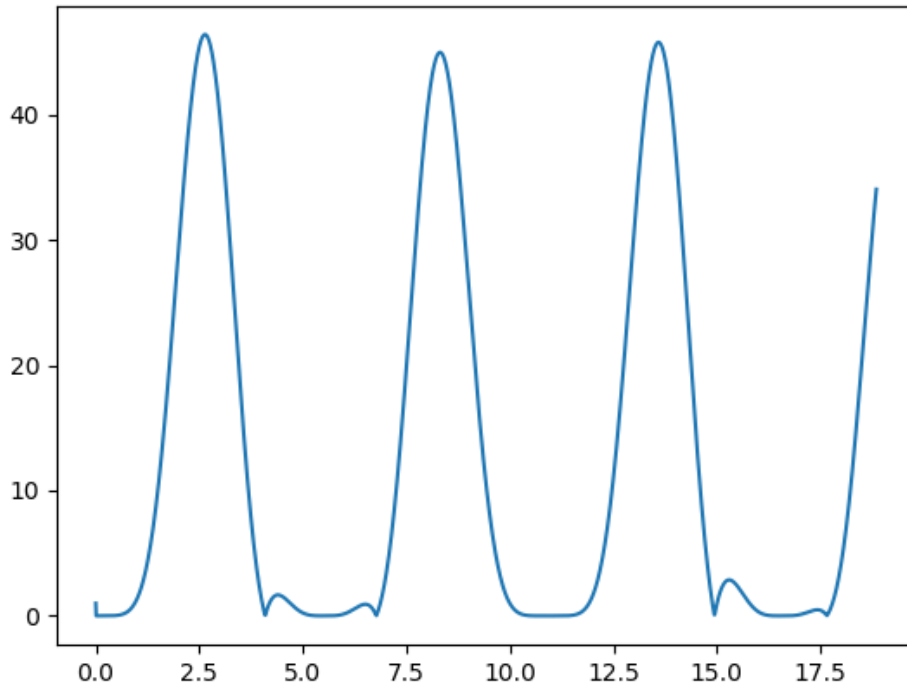


FIGURE 16 – Valeur absolue du membre gauche de l'équation séculaire en fonction de nombre d'onde  $k$ .

On voit donc qu'en fait l'algorithme retourne une fausse valeur propre pour le graphe initial (le graphe équilatéral), qui provient d'un rebond autour de  $(4.14\pi)$  dû aux erreurs de calculs introduites par les inversions matricielles et les exponentielles. Par ailleurs, la valeur absolue du membre gauche de l'équation séculaire prend des valeurs quasi-nulles au voisinage de  $(5.5\pi)^2$  (figure 17), on peut donc légitimement conjecturer que la bonne valeur propre maximale est en fait  $(5.5\pi)^2$ , correspondant à une géométrie équilatérale du graphe.

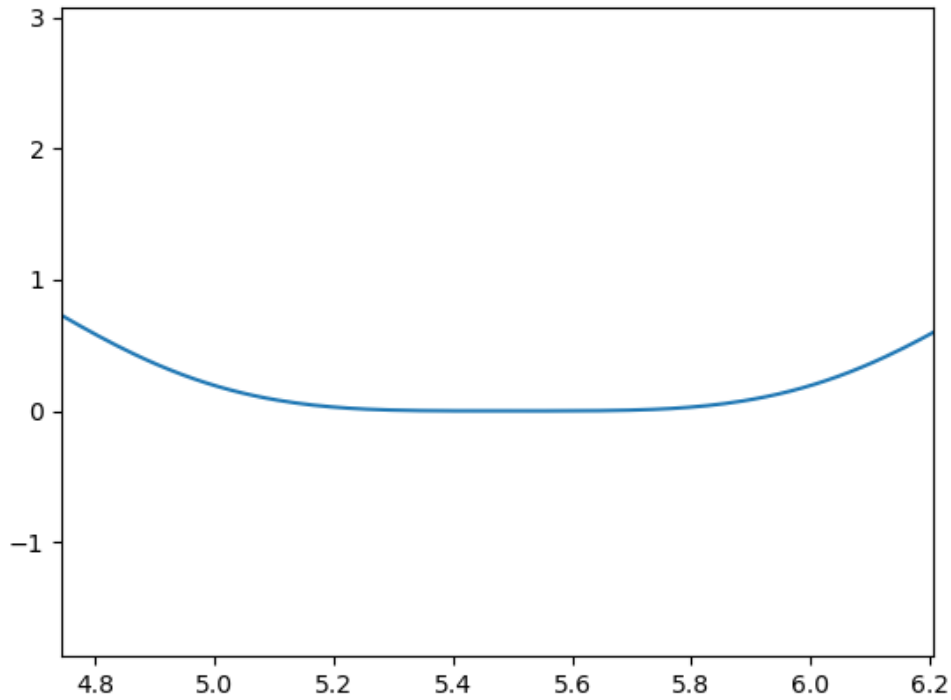


FIGURE 17 – Valeurs prises au voisinage de  $5.5\pi$  par la valeur absolue du membre gauche de l'équation séculaire d'un graphe équilatéral mandarine à 5 arêtes auquel on a adjoint une boucle.

### Deuxième cas : rajout de deux boucles (figure 18).

En contractant l'une des arêtes reliant les deux sommets du graphe, on peut voir que le maximum de la première valeur propre du graphe considéré est au moins  $(6\pi)^2$  (celui d'un bouquet équilatéral à 6 pétales). Un théorème de [6] nous permet d'affirmer que ce maximum ne peut dépasser  $(7\pi)^2$ .

```
>>> (executing file "main.py")
Valeur propre initiale trouvée :5.999999233552276pi
...
Warning: Maximum number of function evaluations has been exceeded.
Valeur propre optimale trouvée :6.05268922558313pi
```

FIGURE 18 – Résultat de la simulation sur un graphe mandarine à 5 arêtes auquel on a rajouté deux boucles.

Grâce à la simulation de la figure 18, on peut voir que la petite valeur propre de ce graphe part de  $(6\pi)^2$  (pour le graphe équilatéral) vers une valeur voisine de  $(6.05\pi)^2$ . On peut donc légitimement conjecturer que le maximum de la première valeur propre pour ce graphe est  $(6\pi)^2$ , correspondant à un graphe équilatéral (figure 19).

```
>>> (executing file "main.py")
Géométrie initiale du graphe :
{0: [(1, 0.16666666666666666), (1, 0.16666666666666666), (1, 0.16666666666666666), (1, 0.16666666666666666), (0, 0.16666666666666666)], 1: [(0, 0.16666666666666666), (0, 0.16666666666666666), (0, 0.16666666666666666), (0, 0.16666666666666666), (0, 0.16666666666666666), (1, 0.16666666666666666)]}
Valeur propre initiale trouvée :5.999999233552276pi
...
Warning: Maximum number of function evaluations has been exceeded.
Valeur propre optimale trouvée :6.05268922558313pi
Géométrie du graphe optimal :
{0: [(1, 0.1696057085260747), (1, 0.16913078307130305), (1, 0.16591612105201614), (1, 0.16519865372746895), (1, 0.16523170130548204), (0, 0.16471369326936308)], 1: [(0, 0.1696057085260747), (0, 0.16913078307130305), (0, 0.16591612105201614), (0, 0.16519865372746895), (0, 0.16523170130548204), (1, 0.16512037136594732)]}
```

FIGURE 19 – Résultat de la simulation sur un graphe mandarine à 5 arêtes auquel on a rajouté deux boucles : les géométries des graphes initial et optimal sont décrites par un dictionnaire de longueurs.

Le fait que la valeur  $(6.05\pi)^2$  obtenue numériquement soit une valeur propre supérieur à celle qu'on conjecture est dû encore une fois au fait que la valeur absolue du membre gauche de l'équation séculaire de ce graphe prend des valeurs quasi-nulles au voisinage de  $6\pi$  (figure 20).

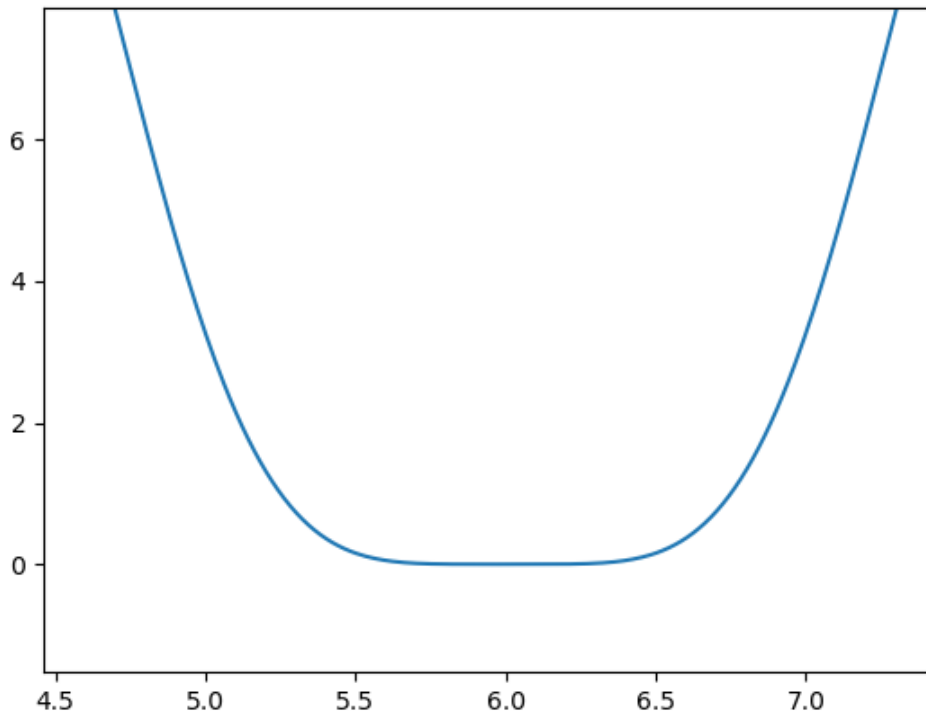


FIGURE 20 – Valeurs prises au voisinage de  $6\pi$  par la valeur absolue du membre gauche de l'équation séculaire d'un graphe équilatéral mandarine à 5 arêtes auquel on a adjoint deux boucles.

### 3.2.5 • EXPLORATION NUMÉRIQUE DE CAS PLUS GÉNÉRAUX

On peut continuer notre exploration numérique pour des cas plus généraux, comme pour par exemple :

- Un graphe mandarine à  $n$  arêtes auquel on a adjoint  $m$  boucles sur l'un des deux sommets (le graphe noté  $G_{nMmB}$  dans le code en annexe).

- Un graphe mandarine à  $n$  arêtes auquel on a adjoint  $m$  feuilles sur l'un des deux sommets (le graphe noté  $G_{nMmA}$  dans le code en annexe).

En plus des problèmes ouverts donnés en exemple dans la section précédente, on peut explorer d'autres graphes appartenant à la première généralisation, comme par exemple le graphe  $G_{7M3B}$  (figure 21) :

```
>>> (executing file "main.py")
Géométrie initiale du graphe :
{0: [(1, 0.11764705882352941), (1, 0.11764705882352941), (1, 0.11764705882352941), (1, 0.11764705882352941), (1, 0.11764705882352941), (1, 0.11764705882352941), (1, 0.11764705882352941)], 1: [(0, 0.11764705882352941), (0, 0.11764705882352941), (0, 0.11764705882352941), (0, 0.11764705882352941), (0, 0.11764705882352941), (0, 0.11764705882352941), (0, 0.11764705882352941)], 2: [(0, 0.11764705882352941), (0, 0.11764705882352941), (0, 0.11764705882352941), (0, 0.11764705882352941), (0, 0.11764705882352941), (0, 0.11764705882352941), (0, 0.11764705882352941)]}
Valeur propre initiale trouvée : 8.500000015187704pi
...
Optimization terminated successfully.
    Current function value: -26.860666
    Iterations: 691
    Function evaluations: 1431
Valeur propre optimale trouvée : 8.550015691684774pi
Géométrie du graphe optimal :
{0: [(1, 0.12102595970504469), (1, 0.11874115958229506), (1, 0.11695907775454821), (1, 0.11695885934154548), (1, 0.11695884806860574), (1, 0.11695887532635149), (1, 0.11695885036311747)], 1: [(0, 0.12102595970504469), (0, 0.11874115958229506), (0, 0.11695907775454821), (0, 0.11695885934154548), (0, 0.11695884806860574), (0, 0.11695887532635149), (0, 0.11695885036311747)], 2: [(0, 0.12102595970504469), (0, 0.11874115958229506), (0, 0.11695907775454821), (0, 0.11695885934154548), (0, 0.11695884806860574), (0, 0.11695887532635149), (0, 0.11695885036311747)]}
```

FIGURE 21 – Résultat de la simulation sur un graphe mandarine à 7 arêtes auquel on a rajouté trois boucles : les géométries des graphes initial et optimal sont décrites par un dictionnaire de longueurs.

Là encore, l'écart entre la valeur optimale obtenue (voisine de  $(8.55\pi)^2$ ) et la valeur conjecturée  $((8.5\pi)^2)$  peut être justifié par la quasi-nullité de la valeur absolue du membre gauche de l'équation séculaire au voisinage de  $8.5\pi$ .

Par ailleurs, en explorant numériquement plusieurs exemples non inclus dans ce rapport, on peut émettre la conjecture suivante :

**Conjecture 3.1.** *Le maximum de la première valeur propre non nulle pour un graphe mandarine à  $n$  arêtes auquel on a adjoint une boucle sur l'un des deux sommets est  $((n + 1/2)\pi)^2$ , valeur propre correspondante à une géométrie équilatérale du graphe considéré.*

Pour la deuxième généralisation, on peut obtenir la conjecture suivante, d'apparence synthétique :

**Conjecture 3.2.** *Le maximum de la première valeur propre non nulle pour un graphe mandarine à  $n$  arêtes auquel on a adjoint  $m$  feuilles sur l'un des deux sommets est  $((n + m/2)\pi)^2$ , valeur propre correspondante à une géométrie équilatérale du graphe considéré.*

On peut justifier cette conjecture en opérant les tests de la figure 23 sur les deux graphes de la figure 22 :

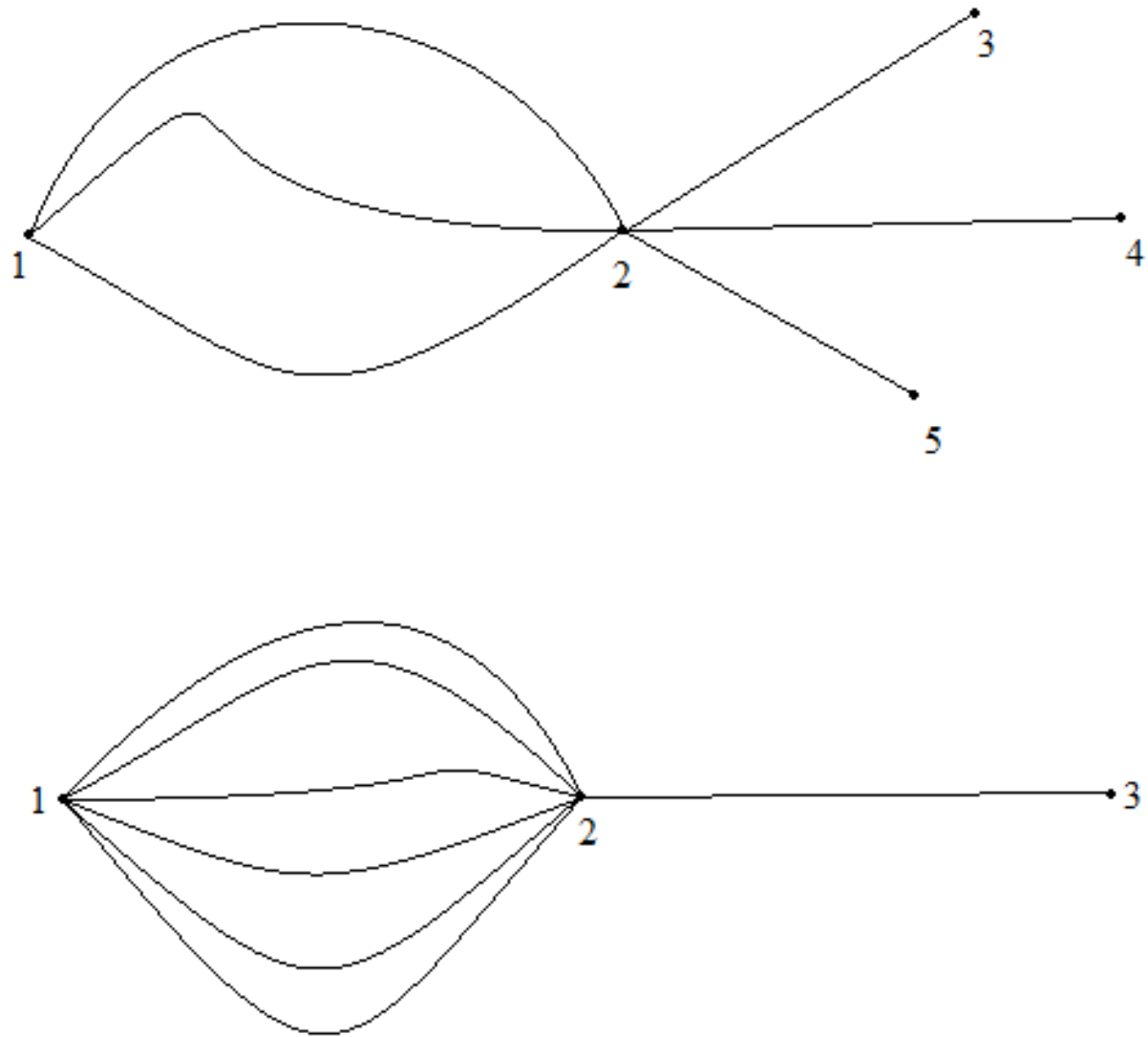


FIGURE 22 – Graphes 3M3A et 6M1A.

```
>>> (executing file "main.py")
Géométrie initiale du graphe :
{0: [(1, 0.125), (1, 0.125), (1, 0.25)], 1: [(0, 0.125), (0, 0.125), (0, 0.25), (2, 0.25), (3, 0.125), (4, 0.125)], 2: [(1, 0.25)], 3: [(1, 0.125)], 4: [(1, 0.125)]}
Valeur propre initiale trouvée :4.000000080800734pi
...
Warning: Maximum number of function evaluations has been exceeded.
Valeur propre optimale trouvée :5.960698027627366pi
Géométrie du graphe optimal :
{0: [(1, 0.134783858438724), (1, 0.16936033729516556), (1, 0.21958672042132613)], 1: [(0, 0.134783858438724), (0, 0.16936033729516556), (0, 0.21958672042132613), (2, 0.21681402115935872), (3, 0.1244462382602717), (4, 0.13500882442515394)], 2: [(1, 0.21681402115935872)], 3: [(1, 0.1244462382602717)], 4: [(1, 0.13500882442515394)]}
```

FIGURE 23 – Résultat de la simulation sur un graphe mandarine à 3 arêtes auquel on a rajouté trois feuilles : les géométries des graphes initial et optimal sont décrites par un dictionnaire de longueurs.

```
>>> (executing file "main.py")
Géométrie initiale du graphe :
{0: [(1, 0.14285714285714285), (1, 0.14285714285714285), (1, 0.14285714285714285), (1, 0.14285714285714285), (1, 0.14285714285714285)], 1: [(0, 0.14285714285714285), (0, 0.14285714285714285), (0, 0.14285714285714285), (0, 0.14285714285714285), (0, 0.14285714285714285), (0, 0.14285714285714285), (2, 0.14285714285714285)], 2: [(1, 0.14285714285714285)]}
Valeur propre initiale trouvée :6.999998257426142pi
...
Optimization terminated successfully.
Current function value: -21.991145
Iterations: 223
Function evaluations: 579
Valeur propre optimale trouvée :6.9999988351403015pi
Géométrie du graphe optimal :
{0: [(1, 0.14277483746790032), (1, 0.14283782777534285), (1, 0.1428062827617647), (1, 0.1427470697894565), (1, 0.1427870532153767), (1, 0.14279002933971593)], 1: [(0, 0.14277483746790032), (0, 0.14283782777534285), (0, 0.1428062827617647), (0, 0.1427470697894565), (0, 0.1427870532153767), (0, 0.14279002933971593), (2, 0.14325689965044308)], 2: [(1, 0.14325689965044308)]}
```

FIGURE 24 – Résultat de la simulation sur un graphe mandarine à 6 arêtes auquel on a rajouté une feuille : les géométries des graphes initial et optimal sont décrites par un dictionnaire de longueurs.

### 3.3 MÉTHODE PROBABILISTE : RECUIT SIMULÉ

Comme on peut s'en convaincre en effectuant les tests précédents, qui sont d'ailleurs réalisés sur des graphes de tailles très petites, l'optimisation de la géométrie d'un graphe quelconque prend énormément de temps lorsqu'on utilise les méthodes précodées dans les bibliothèques standards d'optimisation de Python (`scipy.optimize`). Sans avoir cherché d'autres raisons, implémenter d'autres techniques plus rapides est donc impératif. Toutefois, d'autres raisons rendent l'optimisation à l'aide des bibliothèques standards de Python moins intéressante, voire inutilisable, parmi lesquelles on peut citer :

- Le fait qu'il est difficile de sélectionner la plus petite valeur propre du graphe parmi toutes les valeurs propres possibles si on ne connaît pas au préalable une borne inférieure à l'ensemble des trous spectraux du graphe. Pour cette même raison il est impossible de déterminer la nième valeur propre du graphe grâce à ce genre d'optimisations.
- Le fait que ces techniques supposent certaines propriétés de régularité en les fonctions à optimiser, propriétés qu'on n'est sûr d'avoir dans notre cas que sous les conditions exposées à la section 2.6.
- Le fait que les optimisations qu'on doit réaliser portent sur des fonctions à un très grand nombre de variables (le nombre de variables est exactement le nombre d'arêtes du graphe, donc en moyenne de l'ordre du carré du nombre de sommets du graphe). Les techniques analytiques - celles utilisées par les bibliothèques standards

de Python- sont généralement délaissées au profit des techniques probabilistes comme celle de la descente du gradient probabiliste ou encore du recuit simulé.

C'est cette dernière qu'on a tenté d'implémenter pour les trois raisons suivantes :

- C'est une technique probabiliste, son temps de calcul est donc indépendant de la dimensionnalité du problème.

- Le rapport de Rayleigh peut être interprété comme l'énergie d'une fonction donnée sur un graphe : il est localement minimal pour les fonctions propres du graphe, il peut donc jouer le rôle de l'énergie d'un état dans un algorithme de recuit simulé.

- En imposant une borne inférieure ou en rajoutant un terme bien choisi à l'énergie de l'état courant dans une simulation en recuit simulé, on peut déterminer les valeurs propres d'un graphe de la plus petite vers les plus grandes.

D'après Wikipédia [5], la description générique d'un algorithme de recuit simulé est la suivante : "Le recuit simulé s'appuie sur l'algorithme de Metropolis-Hastings, qui permet de décrire l'évolution d'un système thermodynamique. Par analogie avec le processus physique, la fonction à minimiser deviendra l'énergie  $E$  du système. On introduit également un paramètre fictif, la température  $T$  du système. Partant d'un état donné du système, en le modifiant, on obtient un autre état. Soit celui-ci améliore le critère que l'on cherche à optimiser, on dit alors qu'on a fait baisser l'énergie du système, soit celui-ci le dégrade. Si on accepte un état améliorant le critère, on tend ainsi à chercher l'optimum dans le voisinage de l'état de départ. L'acceptation d'un « mauvais » état permet alors d'explorer une plus grande partie de l'espace des états et tend à éviter de s'enfermer trop vite dans la recherche d'un optimum local."

Comme on l'a déjà mentionné, le rapport de Rayleigh d'une fonction définie sur un graphe quantique est le candidat parfait pour le rôle d'énergie, mais avant cela, il faut définir ce qu'est un état. Dans notre cas, puisqu'on veut réaliser une optimisation sur l'ensemble de toutes les fonctions trigonométriques définies sur le graphe (à condition qu'elles vérifient les conditions aux sommets), et sur l'ensemble de toutes les répartitions de longueurs possibles sur les arêtes du graphe (à condition que la somme des longueurs soit égale à 1), un état peut donc être défini par une configuration métrique du graphe, en plus d'une fonction trigonométrique définie sur ce dernier, structure que l'on a choisie de représenter à l'aide de la classe `GraphFunction` (voir la section 4.2 de l'appendice).

La simulation en recuit simulé est réalisée dans le fichier `main RecuitSimulé.py` (voir la section 4.4 de l'appendice), toutefois, des problèmes de convergence persistent même à des époques très tardives de la simulation, chose qui est généralement occasionnée par des paramètres hyperglobaux mal adaptés.

## RÉFÉRENCES

---

- [1] Berkolaiko, G. et Kuchment, P., *Introduction to Quantum Graphs*, [www.ams.org/bookpages/surv-186](http://www.ams.org/bookpages/surv-186), 2013.
- [2] Gnutzmann, S., Smilansky, U., *Numerical minimization of eigenmodes of a membrane with respect to the domain*, ESAIM : Control, Optimisation and Calculus of Variations, 10(3), 315-330, <https://doi.org/10.1051/cocv:2004011>, 2004.
- [3] Berkolaiko, G. et Kuchment, P., *Quantum graphs : Applications to quantum chaos and universal spectral statistics*, Advances in Physics, 55(5-6), 527-625, <https://doi.org/10.1080/00018730600908042>, 2006.
- [4] Band R., et Lévy, G., *Quantum Graphs which Optimize the Spectral Gap*, Annales Henri Poincarée, <http://levyguillaume.perso.math.cnrs.fr/documents/BandLevyGraphs.pdf>, 2017.
- [5] Page wikipédia décrivant les algorithmes de recuit simulé : [https://fr.wikipedia.org/wiki/Recuit\\_simulé](https://fr.wikipedia.org/wiki/Recuit_simulé).
- [6] Kurasov, P. et Naboko, S., *Rayleigh estimates for differential operators on graphs*, Journal of spectral theory, Volume 4, Issue 2, [https://ems-ph.org/journals/show\\_abstract.php?issn=1664-039Xvol=4iss=2rank=1](https://ems-ph.org/journals/show_abstract.php?issn=1664-039Xvol=4iss=2rank=1), 2014.



## 4 APPENDICE

### 4.1 LA CLASSE GRAPH

```
import numpy as np
from numpy.linalg import inv
from numpy.linalg import det
from math import pi, floor
import scipy.optimize as so
import matplotlib.pyplot as plt
import os

# Classe modélisant un graphe :
class Graph:

    def __init__(self):
        self.vertices = []
        self.edges = {}
        self.E = 0
        self.L = 0

# Rajoute un nouveau sommet au graphe :
    def addNewVertice(self):
        label = len(self.vertices)
        self.vertices.append(label)
        self.edges[label] = []
        return label

#Rajoute une arête au graphe :
    def addEdge(self, v1, v2, d=1):
        if v1 in self.vertices and v2 in self.vertices:
            if v1==v2:
                self.edges[v1].append((v2, d))
                self.E += 1
                self.L += d
                return
            self.edges[v1].append((v2, d))
            self.edges[v2].append((v1, d))
            self.E += 1
            self.L += d
        elif v1 not in self.vertices:
            print(str(v1)+"_not_found_in_the_graph!")
        else:
            print(str(v2)+"_not_found_in_the_graph!")

#Normalise à 1 la longueur totale du graphe :
```

```
def normalize(self):
    n = len(self.vertices)
    L = sum([sum([w[1] for w in self.edges[v]]) for v in self.vertices])/2.0
    for v in self.vertices:
        for itt in range(len(self.edges[v])):
            w = self.edges[v].pop(0)
            self.edges[v].append((w[0],w[1]/L))
    self.L = 1
```

# La méthode suivante transforme le graphe en un graphe simple en rajoutant des sommets sur les arêtes multiples ou les boucles aux sommets :

```
def resolve(self):
    resolved = Graph()
    n = len(self.vertices)
```

# Ajout des sommets du graphe original :

```
for v in range(n):
    resolved.addNewVertice()
```

```
for x in range(n):
```

# Traitement du cas des arêtes multiples :

```
for y in range(x+1,n):
    Exy = [e for e in self.edges[x] if e[0] == y]
    Nxy = len(Exy)
    if Nxy>0:
        for itt in range(1,Nxy):
            e = resolved.addNewVertice()
            d = Exy[itt][1]
            resolved.addEdge(x,e,d/2)
            resolved.addEdge(e,y,d/2)
        d = Exy[0][1]
        resolved.addEdge(x,y,d)
```

# Traitement du cas des sommets à boucles :

```
Exx = [e for e in self.edges[x] if e[0] == x]
Nxx = len(Exx)
if Nxx>0:
    for itt in range(Nxx):
        e = resolved.addNewVertice()
        f = resolved.addNewVertice()
        d = Exx[itt][1]
        resolved.addEdge(x,e,d/3)
        resolved.addEdge(e,f,d/3)
        resolved.addEdge(f,x,d/3)
```

```
return resolved
```

# Matrices de continuité aux sommets du graphe :

```
def A(self,a):
    resolved = self.resolve()
    A = {}
    for v in resolved.vertices:
```

```

        d = len(resolved.edges[v])
        tmp=np.zeros((d,d))
        tmp[-1][0]=np.sin(a)
        tmp[-1][-1]=1
        A[v]=np.eye(d)-np.diagflat((d-1)*[1],1)-tmp
    return A
def B(self,a):
    resolved = self.resolve()
    B = {}
    for v in resolved.vertices:
        d = len(resolved.edges[v])
        tmp=np.zeros((d,d))
        tmp[-1][0]=np.sin(a)
        tmp[-1][-1]=1
        tmp=np.cos(a)*np.array(d*[1])
        B[v]=np.zeros((d,d))
        B[v][-1]=tmp
    return B
def sigma(self,x,a):
    A = self.A(a)
    B = self.B(a)
    sigma = {}
    i=complex(0,1)
    for v in A.keys():
        if det(A[v]+i*x*B[v]) != 0:
            sigma[v] = -inv(A[v]+i*x*B[v]).dot(A[v]-i*x*B[v])
        else:
            sigma[v] = 0*A[v]
    return sigma

# Retourne l'ensemble des arêtes orientées du graphe :
def bonds(self):
    resolved = self.resolve()
    bonds = []
    for v in resolved.vertices:
        for w in resolved.edges[v]:
            bonds.append((v,w[0],w[1]))
    return bonds

# Matrice de diffusion du graphe :
def S(self,x=pi,a=0):
    resolved = self.resolve()
    bonds = self.bonds()
    B = len(bonds)
    S = np.full((B,B),complex(0,0))
    sigma = self.sigma(x,a)
    for b1 in range(B):
        for b2 in range(B):
            if bonds[b1][1] == bonds[b2][0]:
                v = bonds[b1][1]
                c = resolved.edges[v].index((bonds[b1][0],bonds[b1][2]))
                l = resolved.edges[v].index((bonds[b2][1],bonds[b2][2]))
                S[b1][b2] = sigma[v][c][l]

```

**return S**

```
def M(self ,x=pi ,a=0):
    i=complex(0,1)
    bonds = self.bonds()
    B = len(bonds)
    L = [b[2] for b in bonds]
    ExpL=np.diag(np.exp(-i*x*np.array(L)))
    I = np.eye(B)
    return I-self.S(x,a).dot(ExpL)
```

# Membre gauche de l'équation séculaire du graphe :

```
def secEq(self ,x=pi ,a=0):
    return det(self.M(x,a))
```

# Première valeur propre non nulle du graphe :

```
def k1(self ,a=0):
    start = 0
    end = 10*pi*self.E/self.L
    def absSecEq(x):
        return abs(self.secEq(x,a))
    ret = [float("inf")]
    f = 20
    for k in range(1,int(f)):
        retk=so.fminbound(absSecEq ,start+k*(end-start)/f ,
            start+(k+1)*(end-start)/f ,full_output=True)
        if retk[1]<1e-4:
            ret = retk
            break
    return ret[0]
```

# Optimisation de la géométrie du graphe : minimisation de la première valeur propre

```
def minimizeOverGeo(self):
    print("...")
    def f(b):
        G = Graph()
        for v in self.vertices:
            G.addNewVertice()
        i = 0
        for v in self.vertices:
            for w in self.edges[v]:
                if v <= w[0]:
                    if b[i] < 0:
                        return float("inf")
                    G.addEdge(v,w[0],b[i])
                    i += 1
        self.vertices = G.vertices
        self.edges = G.edges
        self.normalize()
        return self.k1()
    b0 = []
    for v in self.vertices:
        for w in self.edges[v]:
```

```

        if v <= w[0]:
            b0.append(w[1])
    b0 = np.array(b0)
    ret = so.minimize(f, b0, method='nelder-mead',
options={'xatol': 1e-12, 'disp': True})
    return self.kl()

```

# Optimisation de la géométrie du graphe : maximisation de la première valeur propre

```

def maximizeOverGeo(self):
    print("...")
    def f(b):
        G = Graph()
        for v in self.vertices:
            G.addNewVertice()
        i = 0
        for v in self.vertices:
            for w in self.edges[v]:
                if v <= w[0]:
                    if b[i] < 0:
                        return float("inf")
                    G.addEdge(v,w[0],b[i])
                    i += 1
        self.vertices = G.vertices
        self.edges = G.edges
        self.normalize()
        return -self.kl()
    b0 = []
    for v in self.vertices:
        for w in self.edges[v]:
            if v <= w[0]:
                b0.append(w[1])
    b0 = np.array(b0)
    ret = so.minimize(f, b0, method='nelder-mead',
options={'xatol': 1e-12, 'disp': True})
    return self.kl()

```

# Module du membre gauche de l'équation séculaire en fonction du paramètre x :

```

def plotNormOfLeft(self,a=0):
    def absSecEq(x):
        return abs(self.secEq(pi*x,a))
    start = 0
    end = self.E*pi/self.L
    X = np.linspace(start,end,num=floor((end-start)*100))
    Y = [absSecEq(x) for x in X]
    plt.clf()
    plt.plot(X,Y)
    plt.show()

```

## 4.2 LA CLASSE GRAPHFUNCTION

```

from math import pi , sin
from copy import deepcopy
import numpy as np
from numpy.linalg import norm
import os
path = 'C:\\Users\\zouha\\Desktop\\Rapport_final_PSC\\PSC.CODE\\'
os.chdir(path)
from Graph import *

# Classe modélisant une fonction sur un graphe :

class GraphFunction:

    def __init__(self,G,epsilon=10**(-4),n=1,a=0):
        self.G = deepcopy(G).resolve()
        self.epsilon = epsilon
        self.n = n
        self.a = a

        self.bonds = self.G.bonds()
        self.B = len(self.bonds)
        self.f = np.full((self.B,1),complex(0,0))

        self.x = self.n*pi
        self.S = self.G.S(self.x,self.a)
        self.I = np.eye(self.B)
        self.index = {}
        for b in range(self.B):
            self.index[(self.bonds[b][0],self.bonds[b][1])] = b

# Transfert des longueurs entre deux arêtes du graphe :

def transfertLengths(self,b1,b2):
    v1b1 = self.bonds[b1][0]
    v2b1 = self.bonds[b1][1]
    v1b2 = self.bonds[b2][0]
    v2b2 = self.bonds[b2][1]

    found = False
    x = None
    for itt in range(len(self.G.edges[v1b1])):
        w = self.G.edges[v1b1].pop(0)
        if not found and w[0] == v2b1:
            x = w
            found = True
            continue
        self.G.edges[v1b1].append(w)
    self.G.edges[v1b1].append((x[0],x[1]-self.epsilon))

    found = False
    x = None
    for itt in range(len(self.G.edges[v2b1])):
        w = self.G.edges[v2b1].pop(0)
        if not found and w[0] == v1b1:

```

```

        x = w
        found = True
        continue
    self.G.edges[v2b1].append(w)
    self.G.edges[v2b1].append((x[0], x[1] - self.epsilon))

    found = False
    x = None
    for itt in range(len(self.G.edges[v1b2])):
        w = self.G.edges[v1b2].pop(0)
        if not found and w[0] == v2b2:
            x = w
            found = True
            continue
        self.G.edges[v1b2].append(w)
        self.G.edges[v1b2].append((x[0], x[1] + self.epsilon))

    found = False
    x = None
    for itt in range(len(self.G.edges[v2b2])):
        w = self.G.edges[v2b2].pop(0)
        if not found and w[0] == v1b2:
            x = w
            found = True
            continue
        self.G.edges[v2b2].append(w)
        self.G.edges[v2b2].append((x[0], x[1] + self.epsilon))

    self.bonds = self.G.bonds()

    return norm((self.G.M()).dot(self.f))

# Incrémentation des coefficients complexes d'une arête :
def update_f(self, b, u, v):
    b_bar = self.index[(self.bonds[b][1], self.bonds[b][0])]
    self.f[b] += u*self.epsilon
    self.f[b_bar] += v*self.epsilon

    return norm((self.G.M()).dot(self.f))

# Incrémentation du paramètre x :
def update_x(self, e):
    self.x += e*self.epsilon

    return norm((self.G.M()).dot(self.f))

# Calcul du rapport de Rayleigh :
def R(self):
    N = 0
    D = 0
    for b in self.bonds:
        L_b = b[2]
        a_b = self.f[self.index[(b[0], b[1])]][0]

```

```

        a_b_bar = self.f[self.index[(b[1],b[0])]][0]
        N += ((abs(a_b)**2)*L_b*self.x)
        -((a_b*np.conjugate(a_b_bar))*sin(L_b*self.x))
        D += ((abs(a_b)**2)*L_b*self.x)
        +((a_b*np.conjugate(a_b_bar))*sin(L_b*self.x))
    if D != 0:
        return abs(N/D)*(self.x)**2
    else:
        return float("-inf")

"""

```

# Jacobien de la fonction qui associe à une géométrie une valeur propre :

```

def jac(self):
    N = 0
    for b in self.bonds:
        L_b = b[2]
        a_b = self.f[self.index[(b[0],b[1])]][0]
        a_b_bar = self.f[self.index[(b[1],b[0])]][0]
        N += ((abs(a_b)**2)*L_b*self.x)
        -((a_b*np.conjugate(a_b_bar))*sin(L_b*self.x))
    return -abs(N)*(self.x)**2

"""

```

## 4.3 LE MAIN

```

import os
path = 'C:\\Users\\zouha\\Desktop\\Rapport_final_PSC\\PSC.CODE\\'
os.chdir(path)
from Graph import *

#Optimisation directe de la géometrie d'un graphe :
#Graphe mandarine à 5 arêtes et à une boucle :

G_5M1B = Graph()
G_5M1B.addNewVertice()
G_5M1B.addNewVertice()
G_5M1B.addEdge(0,1)
G_5M1B.addEdge(0,1)
G_5M1B.addEdge(0,1)
G_5M1B.addEdge(0,1)
G_5M1B.addEdge(0,1)
G_5M1B.addEdge(1,1)
G_5M1B.normalize()

print("Geom trie_initiale_du_graphe:")

print(G_5M1B.edges)

print("Valeur_propre_initiale_trouvée:"+str(G_5M1B.k1()/pi)+"pi")
# Résultat de la simulation :

```



```

print ("Valeur_propre_optimale_trouvée : "+str(G_5M1B.maximizeOverGeo()/pi)+" pi")

print ("Geometrie_du_graphe_optimal : ")

print (G_5M1B.edges)
#Graphe mandarine à 5 arêtes et à deux boucles :
G_5M2B = Graph()
G_5M2B.addNewVertice()
G_5M2B.addNewVertice()
G_5M2B.addEdge(0,1)
G_5M2B.addEdge(0,1)
G_5M2B.addEdge(0,1)
G_5M2B.addEdge(0,1)
G_5M2B.addEdge(0,1)
G_5M2B.addEdge(1,1)
G_5M2B.addEdge(0,0)
G_5M2B.normalize()

print ("Geometrie_initiale_du_graphe : ")

print (G_5M2B.edges)

print ("Valeur_propre_initiale_trouvée : "+str(G_5M2B.k1()/pi)+" pi")
# Résultat de la simulation :
print ("Valeur_propre_optimale_trouvée : "+str(G_5M2B.maximizeOverGeo()/pi)+" pi")

print ("Geometrie_du_graphe_optimal : ")

print (G_5M2B.edges)
#Graphe mandarine à 7 arêtes et à trois boucles :
G_7M3B = Graph()
G_7M3B.addNewVertice()
G_7M3B.addNewVertice()
G_7M3B.addEdge(0,1)
G_7M3B.addEdge(0,1)
G_7M3B.addEdge(0,1)
G_7M3B.addEdge(0,1)
G_7M3B.addEdge(0,1)
G_7M3B.addEdge(0,1)
G_7M3B.addEdge(0,1)
G_7M3B.addEdge(0,1)
G_7M3B.addEdge(1,1)
G_7M3B.addEdge(1,1)
G_7M3B.addEdge(1,1)
G_7M3B.normalize()

print ("Geometrie_initiale_du_graphe : ")

print (G_7M3B.edges)

print ("Valeur_propre_initiale_trouvée : "+str(G_7M3B.k1()/pi)+" pi")

```

```
# Résultat de la simulation :
print ("Valeur_propre_optimale_trouvée : "+str(G_7M3B.maximizeOverGeo()/pi)+"pi")

print ("Geométrie_du_graphe_optimal :")

print (G_7M3B.edges)

#Graphe mandarine à 6 arêtes et à une feuille :
G_6M1A = Graph()
G_6M1A.addNewVertice()
G_6M1A.addNewVertice()
G_6M1A.addNewVertice()
G_6M1A.addEdge(0,1)
G_6M1A.addEdge(0,1)
G_6M1A.addEdge(0,1)
G_6M1A.addEdge(0,1)
G_6M1A.addEdge(0,1)
G_6M1A.addEdge(0,1)
G_6M1A.addEdge(1,2)
G_6M1A.normalize()

print ("Geométrie_initiale_du_graphe :")

print (G_6M1A.edges)

print ("Valeur_propre_initiale_trouvée : "+str(G_6M1A.k1()/pi)+"pi")

# Résultat de la simulation :
print ("Valeur_propre_optimale_trouvée : "+str(G_6M1A.maximizeOverGeo()/pi)+"pi")

print ("Geométrie_du_graphe_optimal :")

print (G_6M1A.edges)

#Graphe mandarine à 3 arêtes et 3 feuilles :
G_3M3A = Graph()
G_3M3A.addNewVertice()
G_3M3A.addNewVertice()
G_3M3A.addNewVertice()
G_3M3A.addNewVertice()
G_3M3A.addNewVertice()
G_3M3A.addEdge(0,1)
G_3M3A.addEdge(0,1)
G_3M3A.addEdge(0,1,2)
G_3M3A.addEdge(1,2,2)
G_3M3A.addEdge(1,3)
G_3M3A.addEdge(1,4)
G_3M3A.normalize()

print ("Geométrie_initiale_du_graphe :")

print (G_3M3A.edges)
```

```
print("Valeur_propre_initiale_trouvée : "+str(G_3M3A.k1()/pi)+"pi")
# Résultat de la simulation :
print("Valeur_propre_optimale_trouvée : "+str(G_3M3A.maximizeOverGeo()/pi)+"pi")

print("Géométrie du graphe optimal :")

print(G_3M3A.edges)
```

## 4.4 LE MAIN DU RECUIT SIMULÉ

```
import os
path = 'C:\\Users\\zouha\\Desktop\\Rapport_final_PSC\\PSC.CODE\\'
os.chdir(path)
from random import uniform
from math import log, exp, floor
from Graph import *
from GraphFunction import *

#Algorithme de recuit simulé pour optimisation de la géométrie d'un graphe :
def h(u):
    return u/(1+u)

# q doit être le plus grand parmi ces trois paramètres afin de permettre à l'énergie de converger vers la valeur
propre de la configuration géométrique "courante"
# Doivent avoir pour somme 1!

p = 0.3
q = 0.5
r = 0.2

# Paramètre de la température du recuit simulé
K = 10

# Seuil de nullité
epsilon = 10**(-4)

# Graphe mandarine a deux boucles
G = Graph()
G.addNewVertice()
G.addNewVertice()
G.addEdge(0,1)
G.addEdge(0,1)
G.addEdge(0,1)
G.addEdge(0,1)
G.addEdge(0,1)
G.addEdge(0,0)
G.addEdge(1,1)
G.normalize()
F = GraphFunction(G)
```

$B = F.B$

"""

*Nomenclature :*

- $T_n$  : *temperature a l'instant  $n$*
- $E$  : *energie de l'etat courant*
- $E_0$  : *plus grande valeur propre parmi celles deja trouvees*
- $w$  : *alea utilise dans les simulations en recuit simule*

"""

$E_0 = 0$

```

for n in range(2,10002):
    print("epoch _=_"+str(n-1))

    Tn = K/log(n)

    U = uniform(0,1)

# Premier cas : on incremente le parametre F.x
    if U < p:

# Transformation de l'etat :
        e = 2*uniform(0,1)-1
        Q = F.update_x(e)

# Alea :
        w = uniform(0,1)

# Énergie :
        E = F.R()

# Test de conformite du nouveau etat :
        if Q >= epsilon or E-E0 < epsilon or w > h(exp(-E/Tn)):

# Retour vers l'etat precedent (de plus faible energie) :
        F.update_x(-e)

# Deuxieme cas : on translate le vecteur F.f suivant un vecteur proportionnel a l'un des vecteurs de la base
canonique (choisis uniformement parmi tous les vecteurs de la base canonique) elif p >= U and U < p+q : #
Transformation de l'etat :
        V = uniform(0,1)
        b = floor(B*V)
        u_Re = 2*uniform(0,1)-1
        u_Im = 2*uniform(0,1)-1
        v_Re = 2*uniform(0,1)-1
        v_Im = 2*uniform(0,1)-1
        u = complex(u_Re,u_Im)
        v = complex(v_Re,v_Im)
        Q = F.update_f(b,u,v)

# Alea :
        w = uniform(0,1)

```

```
# Énergie :
    E = F.R()

# Test de conformité du nouveau état :
    if Q >= epsilon or E-E0 < epsilon or w > h(exp(-E/Tn)):
# Retour vers l'état précédent (de plus faible énergie) :
    F.update_f(b,-u, -v)

# Troisième cas : on transfère une fraction de longueur d'une arête vers l'autre (choisies uniformément et
indépendamment parmi toutes les arêtes du graphe)
    elif p+q <= U :
# Transformation de l'état :
    V1 = uniform(0,1)
    V2 = uniform(0,1)
    b1 = floor(B*V1)
    b2 = floor(B*V2)
    Q = F.transfertLengths(b1,b2)

# Aléa :
    w = uniform(0,1)

# Énergie :
    E = F.R()

# Test de conformité du nouveau état :
    if Q >= epsilon or E-E0 < epsilon or w > h(exp(-F.R()/Tn)):
# Retour vers l'état précédent (de plus faible énergie) :
    F.transfertLengths(b2,b1)

else :
    print("Une_erreur_a_survenue!")
    break

# Résultat de la simulation : print("Résultat de la simulation :") print("Valeur propre trouvée :"+str(F.x))
print("Energie finale (rapport de Rayleigh) :"+str(F.R()))
```