

# Efficient Computation of Optimal Partitions for Spectral Quantities

**Benjamin BOGOSEL**  
 CMAP École Polytechnique  
 Palaiseau, France

benjamin.bogosel@cmap.polytechnique.fr

## Abstract

Recently, a great interest has been shown towards the study of partitions of a fixed domain which optimize quantities related to the first eigenvalue of the Dirichlet Laplace operator associated to each of the cells. There are few situations where exact solutions are known for this type of problems. Therefore, we would like to have algorithms which allow the numerical computation of such optimal partitions. I present below some recent developments which allow us to work on complex partitioning problems without demanding high computational resources. More precisely, I will present results concerning the minimisation of the sum of the eigenvalues

$$\min(\lambda_1(\omega_1) + \dots + \lambda_1(\omega_n))$$

where  $\omega_1, \dots, \omega_n$  represent a partition of a fixed domain  $D$ . Details and examples concerning partitions of domains  $D$  in  $\mathbb{R}^2, \mathbb{R}^3$  and on surfaces in  $\mathbb{R}^3$  will be given.

### 1. Description of the Problem

Given a fixed domain  $D$  in an Euclidean space or in a manifold we say that  $\omega_1, \dots, \omega_n$  form a **partition** of  $D$  if these sets are pairwise disjoint and their union is  $D$ . Let's denote with  $\mathcal{P}_n(D)$  the family of partitions of  $D$  into  $n$  cells. An **optimal partitioning problem** has the form

$$\min_{(\omega_i) \in \mathcal{P}_n(D)} \mathcal{F}(\omega_1, \dots, \omega_n)$$

where  $\mathcal{F}$  is a functional depending on the cells of the partition.

Recall that given a domain  $\omega$  one can consider the spectrum of the Dirichlet-Laplace operator consisting of an increasing sequence of eigenvalues

$$0 < \lambda_1(\omega) \leq \lambda_2(\omega) \leq \dots \leq \lambda_k(\omega) \dots \rightarrow +\infty.$$

For each such eigenvalue  $\lambda_k(\omega)$  there is an associated eigenfunction  $u_k$  which satisfies

$$-\Delta u_k = \lambda_k u_k \text{ in } \omega, \quad u_k = 0 \text{ on } \partial\omega.$$

In the following I will concentrate on the case where  $\mathcal{F}$  is the sum of the fundamental eigenvalues associated to each of the cells:

$$\mathcal{F}(\omega_1, \dots, \omega_n) = \lambda_1(\omega_1) + \dots + \lambda_1(\omega_n).$$

It is a classical result that the optimal partitioning problem

$$\min_{(\omega_i) \in \mathcal{P}_n(D)} \lambda_1(\omega_1) + \dots + \lambda_1(\omega_n) \quad (1)$$

has a solution. Moreover, solutions consist of cells whose boundaries are formed of regular arcs which join at singular points satisfying the equal angle property.

### 2. Motivation and basic properties

The minimization of the sum of the eigenvalues of partitions of a certain fixed domain appears in the following works:

- \* dynamic of systems in competition [Conti, Terracini, Verzini, 05]
- \* modelling of chemical reaction consisting of mutually annihilating reactants [Babin, Cybulska, Holyst, 05]
- \* the study of monotonicity formulas [Alt, Caffarelli, Friedman 80]

Minimizers of (1) have the following properties:

- \* each cell  $\omega_i$  is naturally with non-void. A cell with void interior has  $\lambda_1 = +\infty$ .
- \* each cell  $\omega_i$  is connected

Even if many works address the minimization problem (1) from different points of view, there are almost no cases in which the optimal partition is known explicitly. A famous conjecture due to Caffarelli and Lin says that it is possible that as the number of cells  $n$  goes to  $\infty$  it is the honeycomb partition which solves problem (1).

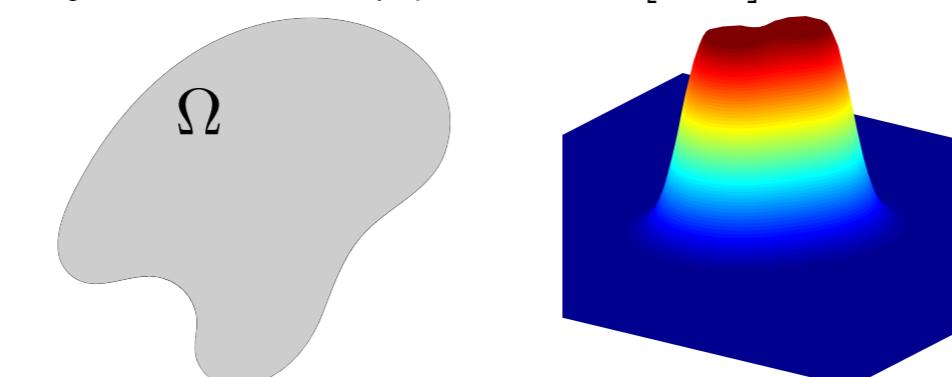
This motivates the development of numerical algorithms. Various algorithms are proposed in the literature. In the following I propose an amelioration of the algorithm introduced in [1]. This amelioration makes possible the numerical computation of solutions of problem (1) for large number of cells in dimension two, on surfaces in  $\mathbb{R}^3$  and in dimension three.

### 3. Numerical Framework

#### Objectives & Challenges

- flexible representation
- move cells without restrictions
- evolution of triple point configurations
- efficient cost function computation
- preserve non-overlapping condition
- compute eigenvalue of each cell
- reasonable simulation time

**Main idea.** Replace each shape  $\omega_i$  by a density function  $\varphi_i : D \rightarrow [0, 1]$ .



#### Advantages

- flexibility, no topology restrictions
- no troubles when moving the cells
- fixed computation grid
- algebraic partition condition:

$$\varphi_1 + \dots + \varphi_n = 1$$

We need to be able to compute the eigenvalue of each cell starting from the functional representation. In order to do this, we consider the following

#### Relaxed Eigenvalue Problem

$$\begin{cases} -\Delta u + \mu u = \lambda(\mu)u & \text{in } D \\ u \in H^1(D) \end{cases}$$

In this way  $\omega$  and the Dirichlet condition are encoded in  $\mu$ :

- $\mu$  large  $\Rightarrow u = 0$
- $\mu = 0 \Rightarrow -\Delta u = \lambda(\mu)u$

Natural choice  $\mu = C(1 - \varphi)$  with  $C \gg 1$ .

**Discretization.** Consider a finite difference grid:  $N \times N$  points on a square. If  $A$  is the discrete finite difference Laplacian on the grid then for each cell we need to solve

$$[A + C \text{diag}(1 - \varphi)]u = \lambda(C, \varphi)u. \quad (2)$$

This can be done using `eigs` in Matlab.

#### Remarks.

- $\lambda_k(C(1 - \chi_\omega)) \rightarrow \lambda_k(\Omega)$  as  $C \rightarrow \infty$
- $\varphi \mapsto \lambda_1(C(1 - \varphi))$  is concave therefore minimizing  $\lambda_1(C(1 - \varphi))$  will make  $\varphi$  a characteristic function

#### Optimization Algorithm (BBO)

Given  $n, D$

- \* start from  $n$  random densities  $\varphi_i : D \rightarrow [0, 1]$  with  $\sum \varphi_i = 1$ .
- \* compute  $\lambda_1(C, \varphi_i)$  and the gradient at each of the grid points (simple expression of the eigenvector...)
- \* Evolve each cell:  

$$\varphi_i \leftarrow \varphi_i - \alpha \nabla \lambda_1(C, \varphi_i)$$
- \* Make sure we still have a partition: projection

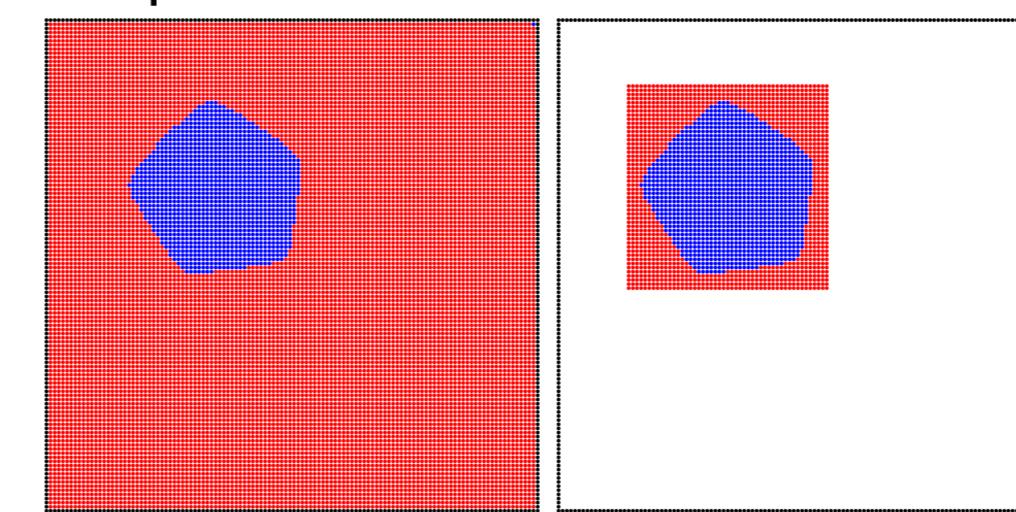
$$\varphi_i \leftarrow \frac{|\varphi_i|}{\sum_i |\varphi_i|}$$

preserves sum condition and concavity!

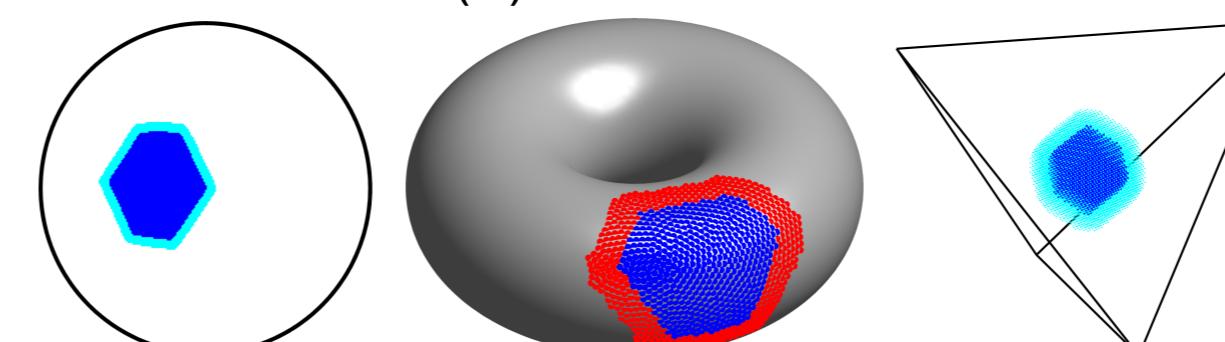
Recall that  $\partial_j \lambda_1(C, \varphi_i) = -C u_j^2$ , where  $u$  is the solution of the matrix eigenvalue problem (2) for  $\varphi = \varphi_i$ .

#### Reduce computational cost.

The original algorithm of Bourdin, Bucur and Oudet computed the relaxed eigenvalues using the whole grid. This is not efficient or necessary. Accuracy and computational costs can be improved by using only points of the grid which are close to the shape.



This procedure allows us to use the algorithm to solve (1) on surfaces or in 3D.



#### New Optimization Algorithm.

Given  $n, D$

1. start from  $n$  random densities  $\varphi_i : D \rightarrow [0, 1]$  with  $\sum \varphi_i = 1$ .
2. restrict the grid to small neighbourhood of  $\{\varphi_i > 0.01\}$
3. compute  $\lambda_1(C, \varphi_i)$  and the gradient on the small sub-grid, using a sub-matrix of the finite difference Laplacian on the whole grid
4. extend gradient to zero outside the small grid
5. Evolve each cell:  

$$\varphi_i \leftarrow \varphi_i - \alpha \nabla \lambda_1(C, \varphi_i)$$
6. Make sure we still have a partition: projection

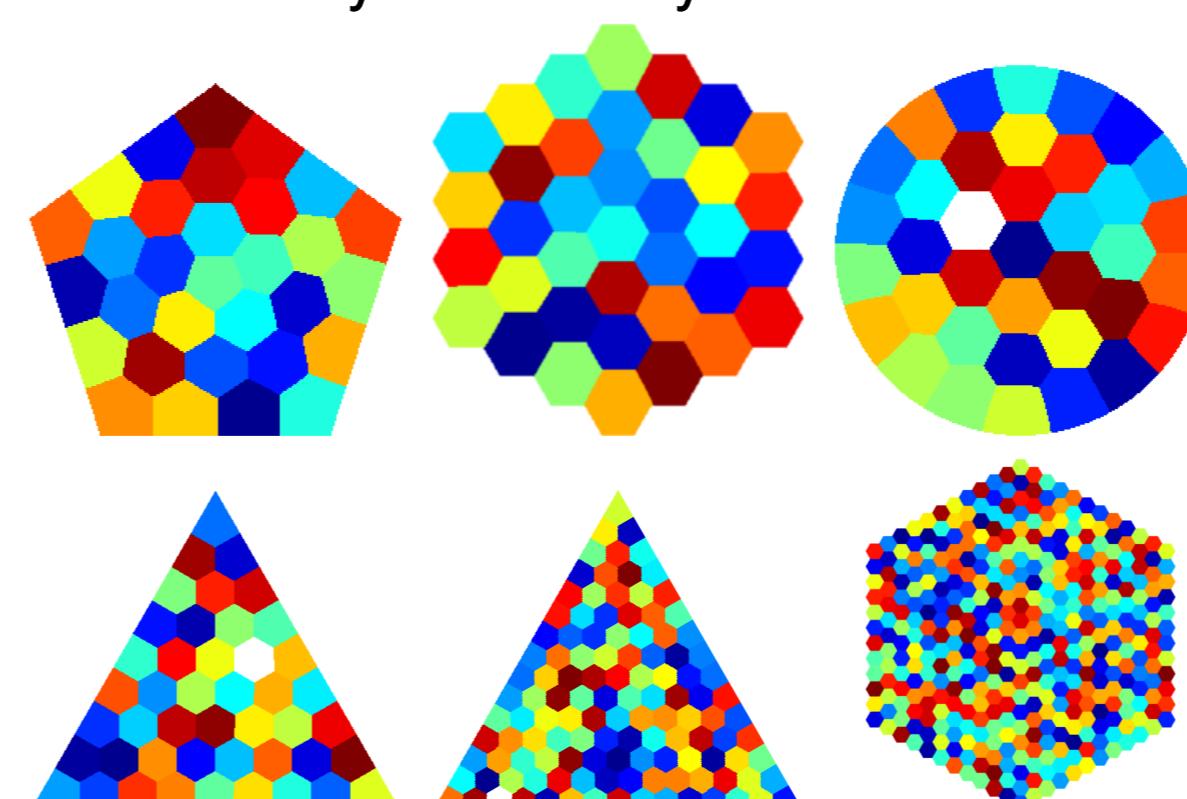
$$\varphi_i \leftarrow \frac{|\varphi_i|}{\sum_i |\varphi_i|}$$

#### Consequences.

- \* dimension of eigenvalue computation matrix is greatly reduced
  - \* larger  $C$ , fewer penalized points  $\rightarrow$  better precision
  - \* huge gain in speed
- New algorithm works well for many cells:
- \* use low resolution until we "see the cells"
  - \* interpolate and continue computations on a finer grid

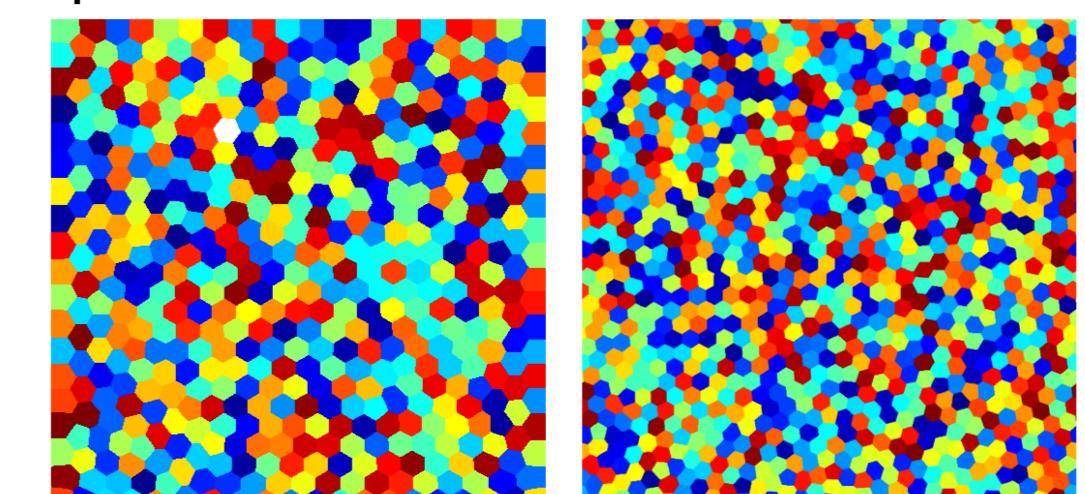
### 4. Numerical simulations

**Very fast for small problems.** The improved algorithm is particularly suited to finding how optimal partitions solving (1) look like for relatively small  $n$ . The results obtained are reliable and local minima are almost always efficiently avoided.



#### Efficient computation for many cells.

Bourdin, Bucur and Oudet used a super computer to find a numerical candidate for  $n = 512$  on a  $512 \times 512$  grid. Using the grid reduction procedure, the same optimization can be done on a laptop. One can even do  $n = 1000$  on a  $1000 \times 1000$  grid with modest resources, in a few hours. Note that in the last case we have an optimization problem with one billion variables.

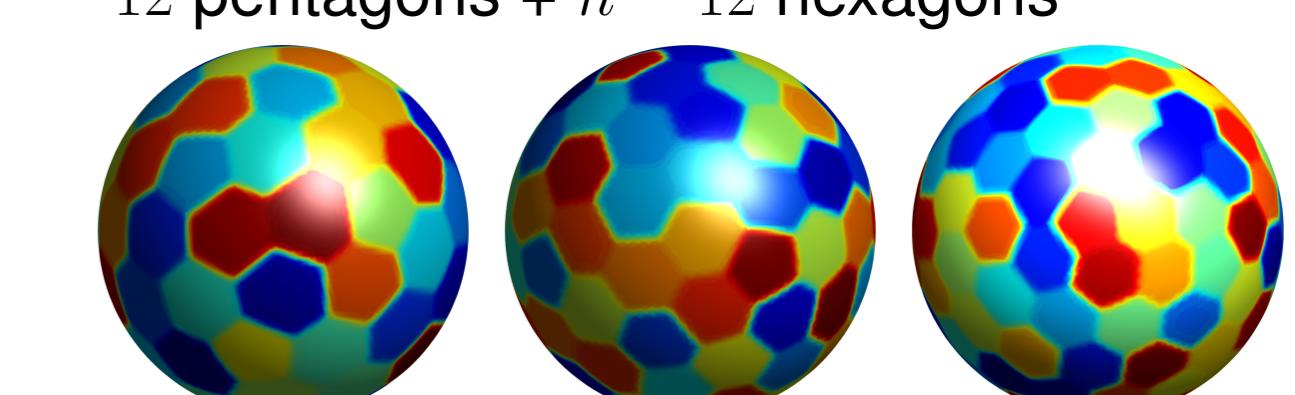


Computations in 2D for many cells confirm the honeycomb conjecture for eigenvalues.

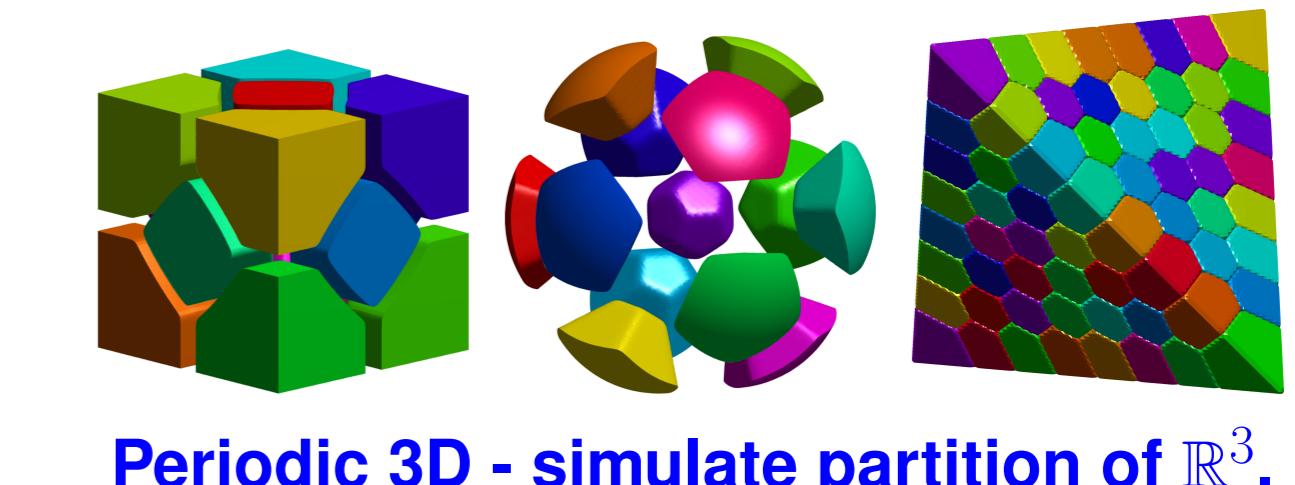
#### Extension to surfaces.

The sphere:

- up to 160000 discretization points
- Computations for  $n \leq 150$
- 12 pentagons +  $n - 12$  hexagons

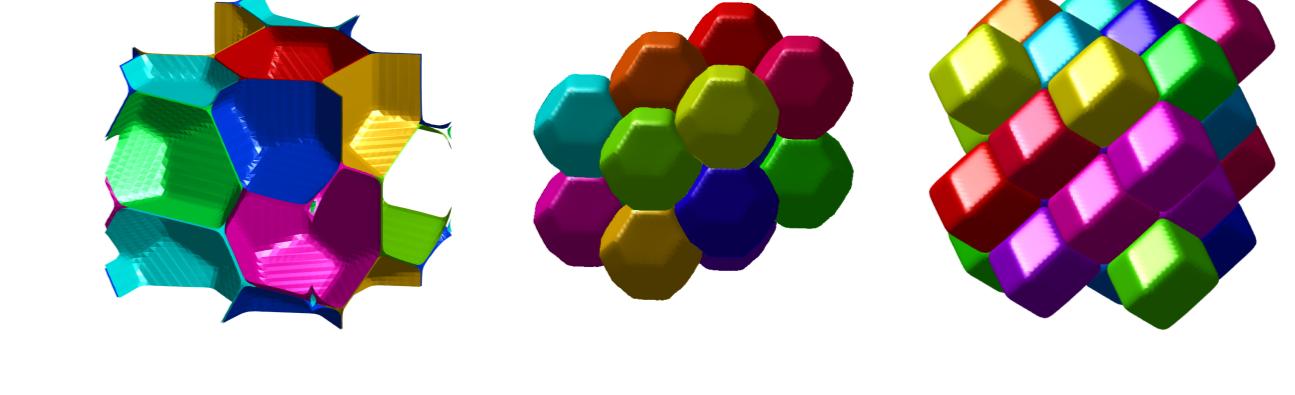


#### 3D computations.



#### Periodic 3D - simulate partition of $\mathbb{R}^3$ .

- \*  $k = 8$  - Weaire-Phelan structure
- \*  $k = 16$  - Kelvin structure
- \*  $k = 32$  - Rhombic dodecahedron tiling



### 5. Perspectives

- handle more complex functionals
- extend the method to other problems which use density representation
- applications



### References

- [1] Bourdin, Blaise and Bucur, Dorin and Oudet, Édouard, Optimal partitions for eigenvalues, SIAM J. Sci. Comput. 2009.
- [2] B. Bogosel, Efficient algorithm for optimizing spectral partitions, 2017.

More details and other various computations on my webpage:

