

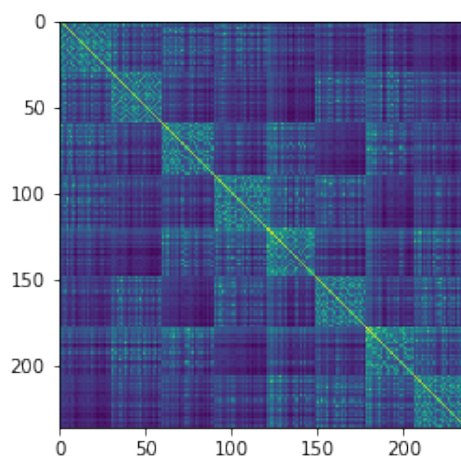


PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
IMT3810 - TÓPICOS AVANZADOS EN ANÁLISIS NUMÉRICO  
PROFESOR: ELWIN VAN'T WOUT

# Proyecto

---

## Matrices Jerárquicas



**Estudiante:** Francisca Muñoz

**Fecha entrega:** 17 de diciembre de 2023

# Contents

<b>1</b>	<b>Introducción y Metodología</b>	<b>3</b>
<b>2</b>	<b>Conceptos importantes</b>	<b>5</b>
2.1	Adaptative Cross Approximation (ACA) . . . . .	5
2.2	Condición de Admisibilidad (Admissibility condition) . . . . .	5
2.3	Operaciones: Matvec . . . . .	6
<b>3</b>	<b>Ejemplo: Single Layer Potential para formulación de Laplace</b>	<b>7</b>
<b>4</b>	<b>Referencias</b>	<b>8</b>

# 1 Introducción y Metodología

Considerando que el método de elementos de frontera "BEM" genera una matriz densa, que necesita mucha memoria para poder guardarse, vamos a desarrollar un método para poder hacer las operaciones de forma más eficiente.

Para esto, utilizando la librería Bempp-c1 realizaremos la compresión jerárquica de dichas matrices. Esta compresión consiste en descomponer la matriz en diferentes bloques. Algunos de estos bloques serán aproximados por aproximaciones de bajo rango y otras serán mantenidas. Esto es posible por la construcción del método Bem y el uso de la función de Green, ya que los nodos más cercanos entre sí tendrán un peso mayor a los lejanos.

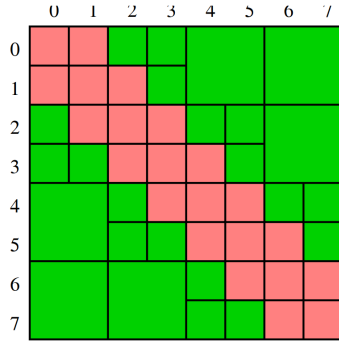


Figure 1: Partición para  $p = 3$

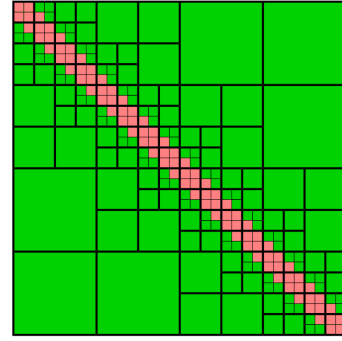


Figure 2: Partición para  $p = 5$

La construcción de la matriz jerárquica o  $\mathcal{H}$ -Matrix se realizará de forma recursiva. Si consideramos que para la primera partición de la matriz  $M \in \mathbb{R}^{2^n \times 2^n}$  obtendremos:

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} =: \mathcal{H}_1$$

Con  $M_{11}, M_{12}, M_{21}, M_{22} \in \mathbb{R}^{2^{n-1} \times 2^{n-1}}$ . Luego a las matrices  $M_{12}$  y  $M_{21}$  se les hará una aproximación de bajo rango utilizando **ACA**, mientras que en las matrices diagonales continuará el algoritmo recursivo. Así, obtenemos la siguiente representación para un operador de matriz en descomposición jerárquica:

$$\mathcal{H}_p = \begin{bmatrix} \mathcal{H}_{p-1} & \mathcal{R}_{p-1} \\ \mathcal{R}_{p-1} & \mathcal{H}_{p-1} \end{bmatrix}$$

Con  $\mathcal{H}_p \in \mathbb{R}^{2^p \times 2^p}$  y  $\mathcal{H}_{p-1}, \mathcal{R}_{p-1} \in \mathbb{R}^{2^{p-1} \times 2^{p-1}}$ . Considerando que en las matrices  $\mathcal{H}$  se continúa con la recursión, y en las matrices  $\mathcal{R}$  se realiza la descomposición de bajo rango. Así obtendremos una compresión de la forma mostrada en las Figuras 1 y 2.

Algunos atributos clave, como el número de bloques en los que se va a descomponer la matriz, el número de bloques de bajo rango, el número de bloques denso y la cantidad de memoria utilizada dependerán también de la forma en que se distribuirán los bloques de sub-matrices. Esto se obtiene a partir de la **condición de admisibilidad**, que más adelante será descrita.

Hay ciertos atributos y/o magnitudes que es necesario tener en cuenta a la hora de realizar esta compresión:

## 1. Número de bloques del operador H-Matrix

Considerando que una matriz de  $1 \times 1$  contiene un bloque, y que en cada paso se agregan dos bloques más el doble de bloques anteriores, para una matriz de  $p \times p$  obtendremos la siguiente expresión:  $N(p) = 2 + 2N(p-1)$ . Obtenemos que para una matriz de  $n \times n$  tendremos un total de  $3n - 2$  bloques.

2. **Número de bloques densos del operador H-Matrix**

Podemos ver que por cada fila, tendremos 3 bloques densos, y en la primera y última fila solo serán 2.

3. **Número de bloques de bajo rango del operador H-Matrix**

4. **Cantidad de memoria utilizada en el operador H-Matrix (kb)**

Primero vemos que las matrices  $\mathcal{R}_p$  tienen un costo de almacenamiento de  $CR(p) = 2^{p+1}$ . Luego, las matrices  $\mathcal{H}_p$ , definidas recursivamente tendrán un costo de almacenamiento de  $CH(p) = 2 \cdot CR(p - 1) + 2 \cdot CH(p - 1)$ , con  $CH(0) = 1$ . Así obtenemos finalmente que el costo total de almacenamiento será:  $CT = 2 + 2n \log_2 n$ , para una matriz de  $n \times n$ .

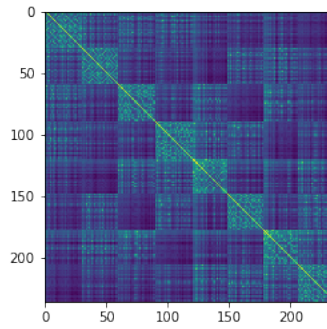


Figure 3: Ejemplo de single layer Potential para Laplace con malla esférica

## 2 Conceptos importantes

### 2.1 Adaptive Cross Approximation (ACA)

Las matrices definidas anteriormente como  $\mathcal{R}_p$ , serán aproximadas por el método de bajo rango "Adaptive cross Approximation". En este método el rango de la aproximación será determinado en función de la submatriz, para disminuir el error convenientemente. Este método permite descomponer la matriz en el producto de dos matrices rectangulares, disminuyendo considerablemente el costo de almacenamiento.

A continuación se presenta un pseudocódigo del algoritmo para matrices cuadradas. El pseudocódigo

---

#### Algorithm 1: Adaptive Cross Approximation

---

**Data:**  $M \in \mathbb{R}^{\hat{I} \times \hat{S}}$ ,  $\tau > 0$  (determina precisión),  $\varepsilon_{abs} > \tau$   
**Result:** Aproximación de rango  $k$  variable  $\sum_{v=1}^k a^v (b^v)^T$

```

1 while  $\varepsilon_{abs} > \tau$  do
2   Se busca la entrada en módulo más grande;
3    $(i_v, j_v) := \operatorname{argmax}_{(i,j)} |M_{i,j}|$ ,  $\delta := |M_{i_v, j_v}|$ ;
4   if  $\delta = 0$  then
5     El algoritmo termina con rango  $v - 1$ ;
6   else
7     Se computan las entradas de  $a^v$  y  $b^v$ ;
8      $(a^v)_i := M_{i, j_v}$ ,  $i \in \hat{I}$ ,  $(b^v)_j := M_{i_v, j} / \delta$ ,  $j \in \hat{S}$ ;
9     Se actualizan los valores de la matriz;
10     $M_{i,j} := M_{i,j} - (a^v)_i (b^v)_j$ ,  $i \in \hat{I}$ ,  $j \in \hat{S}$ ;
11     $\varepsilon_{abs} = \|a^v\|_2 \|b^v\|_2$ ;
12     $v = v + 1$ 
13 return  $a^v, b^v$ 

```

---

anterior, se encuentra en el repositorio entregado. Así, por cada sub-matriz, que cumpla con el criterio de admisibilidad, será descompuesta en el producto de dos matrices rectangulares más pequeñas.

### 2.2 Condición de Admisibilidad (Admissibility condition)

Como se mencionó anteriormente, la condición de admisibilidad nos indicará de que manera se deberá realizar la partición de la matriz original.

A continuación se presentan dos ejemplos clásicos de dichas particiones.

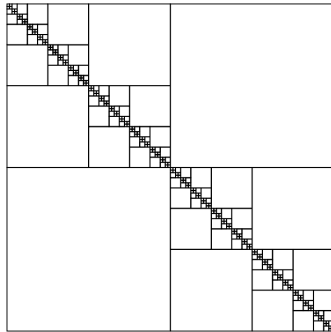


Figure 4: Weakly admissibility condition

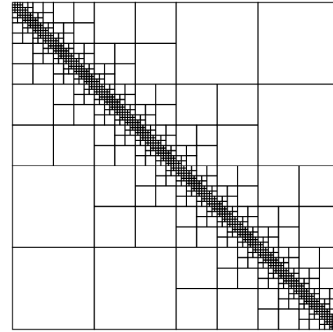


Figure 5: Strongly admissibility condition

La partición utilizada depende también de la dimensionalidad del problema, entre otros aspectos. En las imágenes anteriores vemos dos criterios de partición diferentes, llamados **Condición de admi-**

**bilidad débil** y **Condición de admisibilidad fuerte**, respectivamente. En el primer caso (Figura 4) se ve el criterio de admisibilidad débil, en que únicamente los bloques diagonales se almacenarán como bloques densos, mientras que para todos los otros bloques se realizará una aproximación de bajo rango (ACA). Esta aproximación se realiza de manera recursiva, sobre los bloques que no forman parte de la diagonal de la matriz, quedando así finalmente los bloques diagonales como densos. Por otro lado, la condición de admisibilidad fuerte (Figura 5) es similar a la débil, pero presentando una mayor cantidad de sub-matrices, como se puede apreciar en la figura. En este caso, las matrices densas son las correspondientes a la diagonal y también sus vecinas (Se observa en Figura 1 y Figura 2 igualmente).

### 2.3 Operaciones: Matvec

Para la implementación de la operación **matriz**  $\times$  **vector**, se utilizó el caso específico para la condición débil de admisibilidad. Para esto, la función debe ser definida de forma recursiva sobre los bloques de sub-matrices. A continuación se encuentra la forma de implementación, considerando la siguiente notación:  $(M_{i,j})^k$ ,  $i$  : fila,  $j$  : columna,  $k$  : número de iteración.

$$\begin{pmatrix} M_{1,1}^1 & M_{1,2}^1 \\ M_{2,1}^1 & M_{2,2}^1 \end{pmatrix} \cdot \begin{pmatrix} b_1^1 \\ b_2^1 \end{pmatrix} = \begin{pmatrix} M_{1,1}^1 b_1^1 + M_{1,2}^1 b_2^1 \\ M_{2,1}^1 b_1^1 + M_{2,2}^1 b_2^1 \end{pmatrix}$$

Luego, las operaciones  $M_{1,1}^1 b_1^1$  y  $M_{2,2}^1 b_2^1$  deben realizarse de forma recursiva hasta llegar a las sub-matrices densas. Por otro lado, las operaciones  $M_{1,2}^1 b_2^1$  y  $M_{2,1}^1 b_1^1$  se realizarán, descomponiendo las matrices según la Adaptive Cross Approximation. Así obtenemos :

$$M_{1,2}^1 b_2^1 \simeq A_{2,1}^1 (B_{1,2}^{1T} b_2^1)$$

$$M_{1,2}^1 b_2^1 \simeq A_{1,2}^1 (B_{1,2}^{1T} b_2^1)$$

### 3 Ejemplo: Single Layer Potential para formulación de Laplace

Para comenzar, realizaremos la compresión en matriz jerárquica de el operador "**Single Layer potential**" para Laplace, ya que es el más simple. En el notebook incluido en el repositorio de GitHub se encuentra un ejemplo realizado para una malla esférica y utilizando funciones de base **P1**.

Segun el texto *Hierarchical Matrices Based on a Weak Admissibility Criterion*(W. Hackbusch, B. N. Khoromskij, R. Kriemann), la condición débil de admisibilidad es apropiado para utilizar con el potencial de Single-Layer para el problema de Laplace.

## 4 Referencias

- 1 Betcke, Timo, Elwin van 't Wout, and Pierre G  lat (2017). "Computationally efficient boundary element methods for high-frequency Helmholtz problems in unbounded domains". In: Modern Solvers for Helmholtz Problems. Ed. by Domenico Lahaye, Jok Tang, and Kees Vuik. Geosystems Mathematics. Cham: Birkh  user, pp. 215–243. doi: 10.1007/978-3-319-28832-1\_9.
- 2 B  rm, Steffen (2010). Efficient numerical methods for non-local operators: H2-matrix compression, algorithms and analysis. Vol. 14. EMS Tracts in Mathematics. Z  rich: European Mathematical Society.
- 3 B  rm, Steffen, Lars Grasedyck, and Wolfgang Hackbusch (2003). Hierarchical Matrices. Tech. rep. Leipzig: Max-Planck-Institut f  r Mathematik in den Naturwissenschaften. url: <https://www.mis.mpg.de/publications/other-series/ln/lecturenote-2103.html>
4. <https://link.springer.com/content/pdf/10.1007/s00607-004-0080-4.pdf>