

CIMAT

# Precondicionamiento con factorizaciones incompletas

*Miguel Vargas-Félix*

miguelvargas@ciamat.mx  
<http://www.cimat.mx/~miguelvargas>

# Número de condición

El número de condición  $\kappa$  de una matriz  $\mathbf{A}$  no singular, para una norma  $\|\cdot\|$  está dado por

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\|.$$

Para la norma  $\|\cdot\|_2$ ,

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| = \frac{\sigma_{max}(\mathbf{A})}{\sigma_{min}(\mathbf{A})},$$

donde  $\sigma$  son los valores singulares de la matriz.

Para una matriz  $\mathbf{A}$  simétrica positiva definida,

$$\kappa(\mathbf{A}) = \frac{\lambda_{max}(\mathbf{A})}{\lambda_{min}(\mathbf{A})},$$

donde  $\lambda$  son los eigenvalores de  $\mathbf{A}$ .

El **número de condición** indica que tan sensible es una función a pequeños cambios en la entrada.

Así, matrices con un número de condición cercano a 1 se dicen que están bien condicionadas.

Un sistema de ecuaciones  $\mathbf{A} \mathbf{x} = \mathbf{b}$  es considerado

**bien condicionado** si un pequeño cambio en los valores de  $\mathbf{A}$   
o un pequeño cambio en  $\mathbf{b}$  resulta en un pequeño cambio en  $\mathbf{x}$ .

Un sistema de ecuaciones  $\mathbf{A} \mathbf{x} = \mathbf{b}$  es considerado

**mal condicionado** si un pequeño cambio en los valores de  $\mathbf{A}$   
o un pequeño cambio en  $\mathbf{b}$  resulta en un cambio grande en  $\mathbf{x}$ .

Al reducir el número de condición de un sistema, se puede incrementar la velocidad de convergencia del gradiente conjugado.

# Gradiente conjugado preconditionado

Entonces, en vez de resolver el problema

$$\mathbf{A} \mathbf{x} - \mathbf{b} = 0,$$

se resuelve el problema

$$\mathbf{M}^{-1}(\mathbf{A} \mathbf{x} - \mathbf{b}) = 0,$$

con  $\mathbf{M}^{-1}$  una matriz cuadrada, la cual recibe el nombre de **precondicionador**.

El mejor preconditionador sería claro  $\mathbf{M}^{-1} = \mathbf{A}^{-1}$ , así  $\mathbf{x} = \mathbf{M}^{-1} \mathbf{b}$ , y el gradiente conjugado convergiría en un paso.

Al igual que la matriz  $\mathbf{A}$ , el preconditionador  $\mathbf{M}^{-1}$  tiene que ser simétrico positivo definido.

Hay dos tipos de preconditionadores, implícitos  $\mathbf{M}$  y explícitos  $\mathbf{M}^{-1}$ .

En algunos casos es costoso calcular  $\mathbf{M}^{-1}$ , en general se utilizan preconditionadores con inversas fáciles de calcular o preconditionadores implícitos que se puedan factorizar.

El algoritmo es el siguiente:

entrada:  $\mathbf{A}$ ,  $\mathbf{x}_0$ ,  $\mathbf{b}$ ,  $\varepsilon$

$$\mathbf{r}_0 \leftarrow \mathbf{A} \mathbf{x}_0 - \mathbf{b}$$

$$\mathbf{q}_0 \leftarrow \mathbf{M}^{-1} \mathbf{r}_0$$

$$\mathbf{p}_0 \leftarrow -\mathbf{q}_0$$

$$k \leftarrow 0$$

mientras  $\|\mathbf{r}_k\| > \varepsilon$

$$\mathbf{w} \leftarrow \mathbf{A} \mathbf{p}_k$$

$$\alpha_k \leftarrow \frac{\mathbf{r}_k^T \mathbf{q}_k}{\mathbf{p}_k^T \mathbf{w}}$$

$$\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

$$\mathbf{r}_{k+1} \leftarrow \mathbf{r}_k + \alpha \mathbf{w}$$

$$\mathbf{q}_{k+1} \leftarrow \mathbf{M}^{-1} \mathbf{r}_{k+1}, \text{ o resolver } \mathbf{M} \mathbf{q}_{k+1} \leftarrow \mathbf{r}_{k+1}$$

$$\beta_k \leftarrow \frac{\mathbf{r}_{k+1}^T \mathbf{q}_{k+1}}{\mathbf{r}_k^T \mathbf{q}_k}$$

$$\mathbf{p}_{k+1} \leftarrow -\mathbf{q}_{k+1} + \beta_{k+1} \mathbf{p}_k$$

$$k \leftarrow k+1$$

Nótese que ahora el algoritmo requiere aplicar el preconditionador en cada paso.

# Precondicionador Cholesky incompleto

Éste consiste en tener un preconditionador

$$\mathbf{M} = \mathbf{H}_t \mathbf{D} \mathbf{H}_t^T,$$

dónde  $\mathbf{H}_t$  es una matriz triangular inferior rala que tiene una estructura similar a la factorización Cholesky de  $\mathbf{A}$ . El entero  $t$  indicará el nivel de truncamiento.

La estructura de  $\mathbf{H}_0$  será igual a la estructura de la parte triangular inferior de  $\mathbf{A}$ .

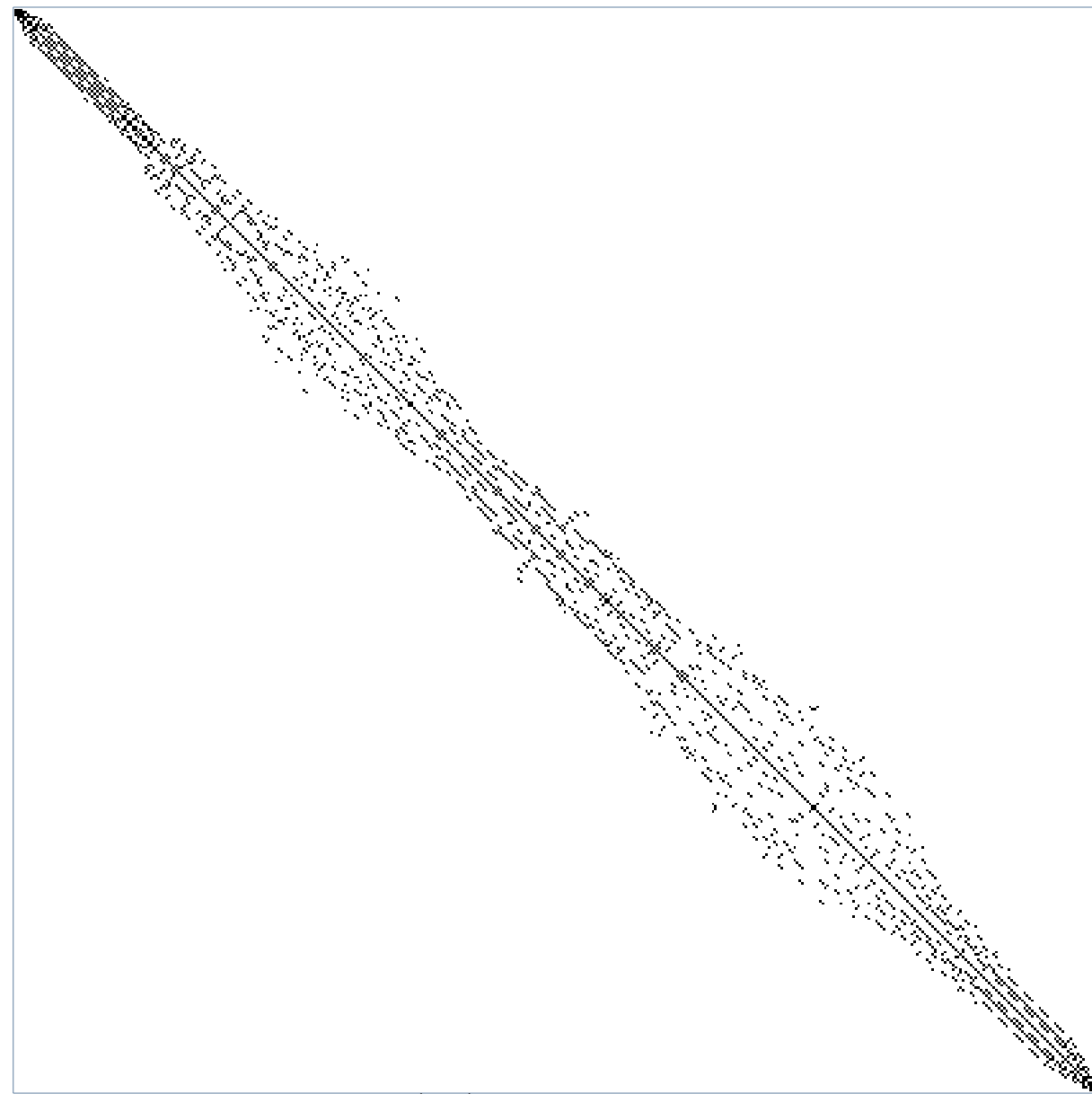
La estructura de  $\mathbf{H}_n$  será igual a la estructura de  $\mathbf{L}$  (factorización Cholesky de  $\mathbf{A}$ ).

Los valores de  $\mathbf{H}_t$  se llenarán utilizando la fórmulas de la factorización Cholesky, así

$$H_{ii} = 1,$$
$$H_{ij} = \frac{1}{D_j} \left( A_{ij} - \sum_{k=1}^{j-1} H_{ik} H_{jk} D_k \right), \text{ para } i > j$$
$$D_j = A_{jj} - \sum_{k=1}^{j-1} H_{jk}^2 D_k.$$

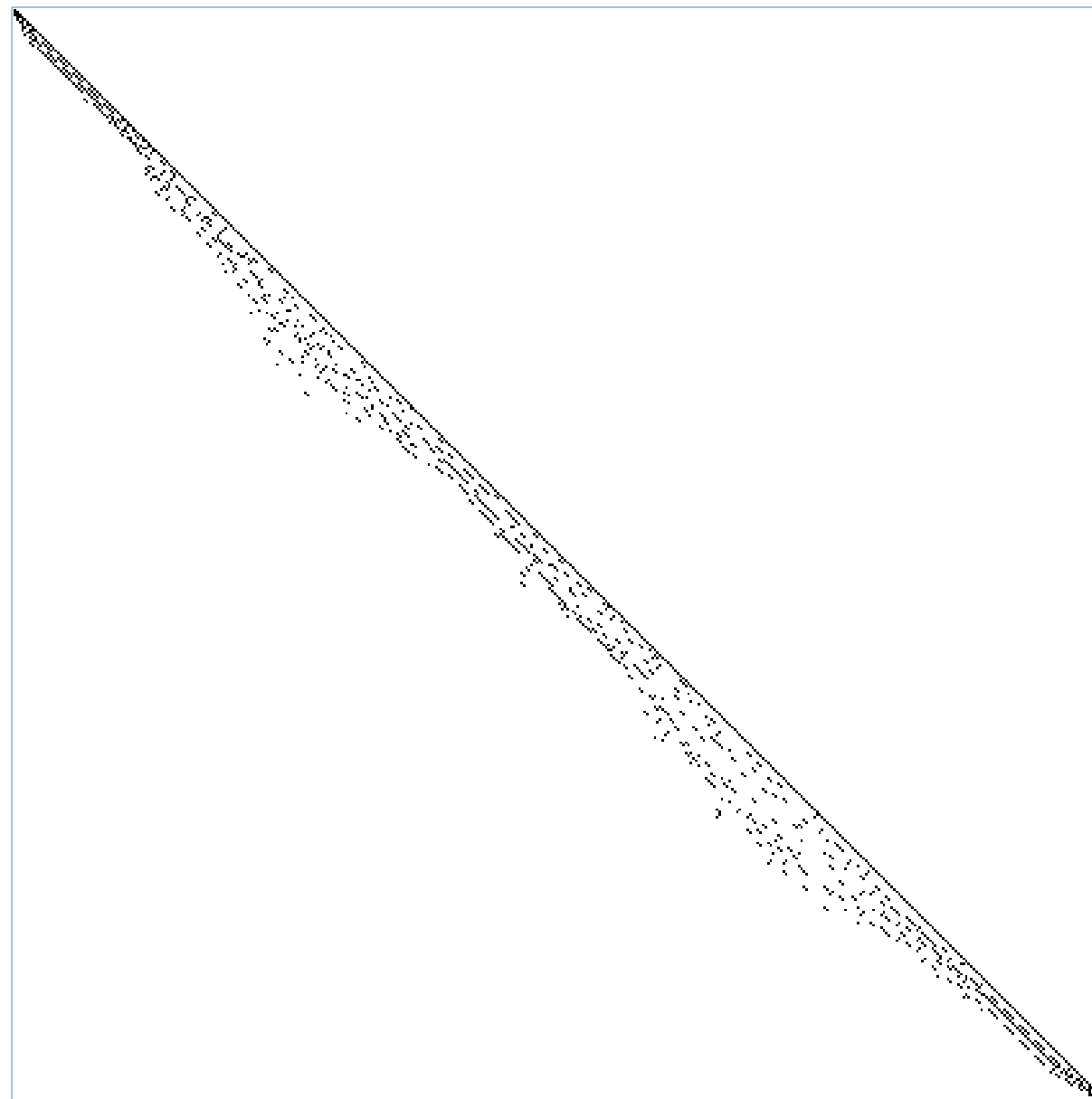
Veamos un ejemplo, para una matriz  $\mathbf{A} \in \mathbb{R}^{424 \times 424}$ , la estructura de  $\mathbf{H}_t$  sería...

$\mathbf{A} =$



$$\eta(\mathbf{A}) = 2456$$

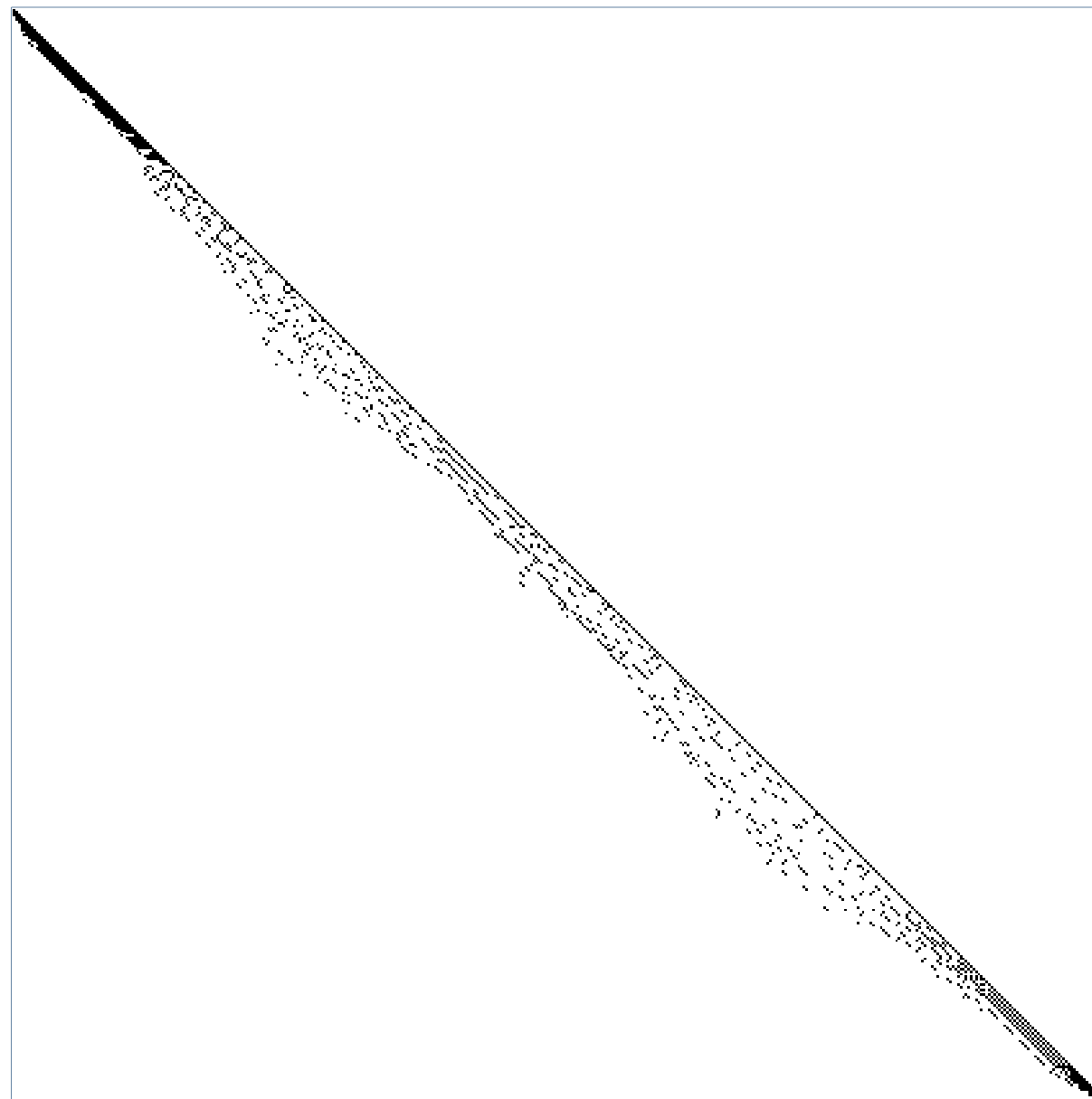
$\mathbf{H}_0 =$



$$\eta(\mathbf{H}_0) = 1436$$

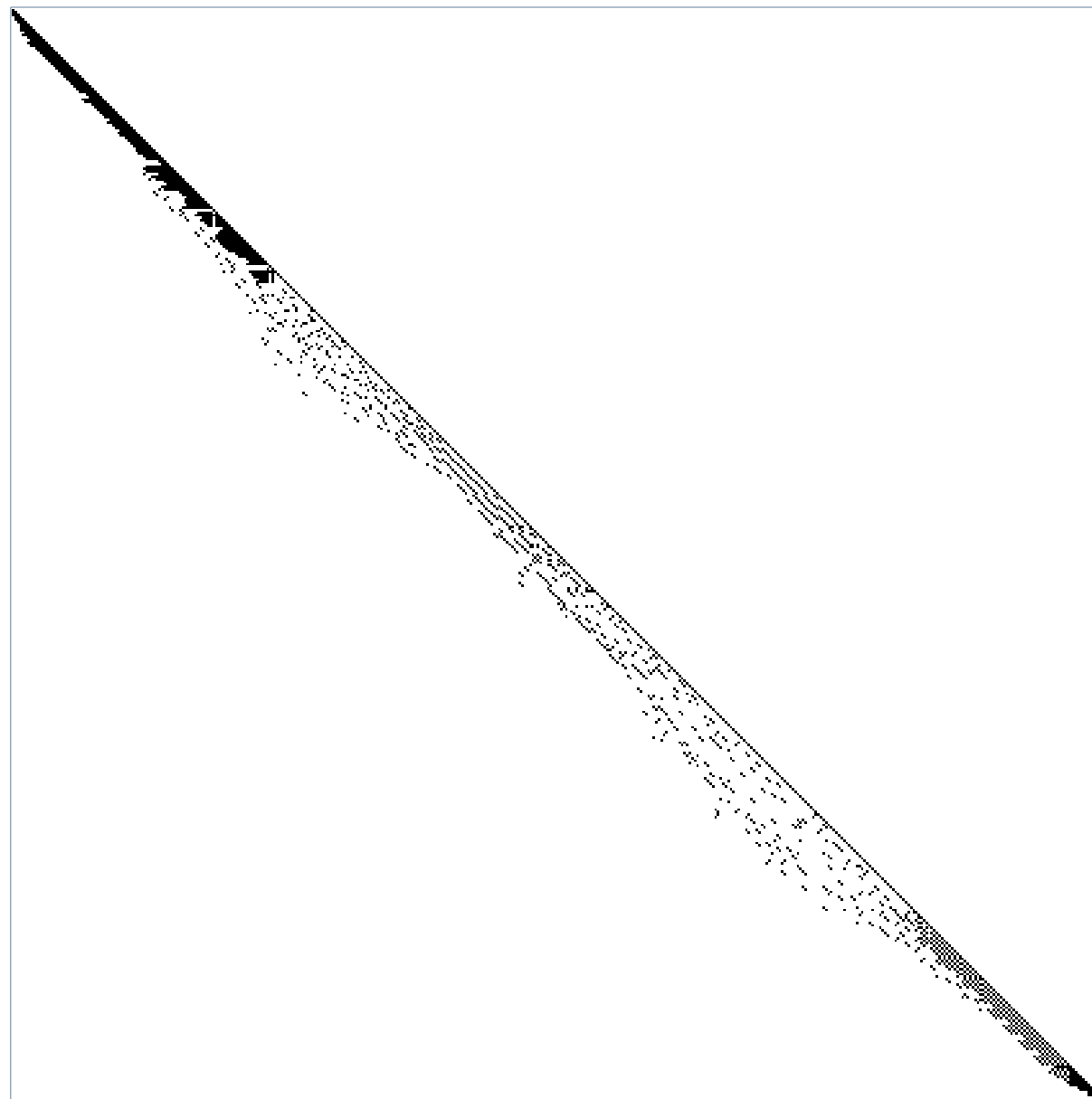


$\mathbf{H}_5 =$



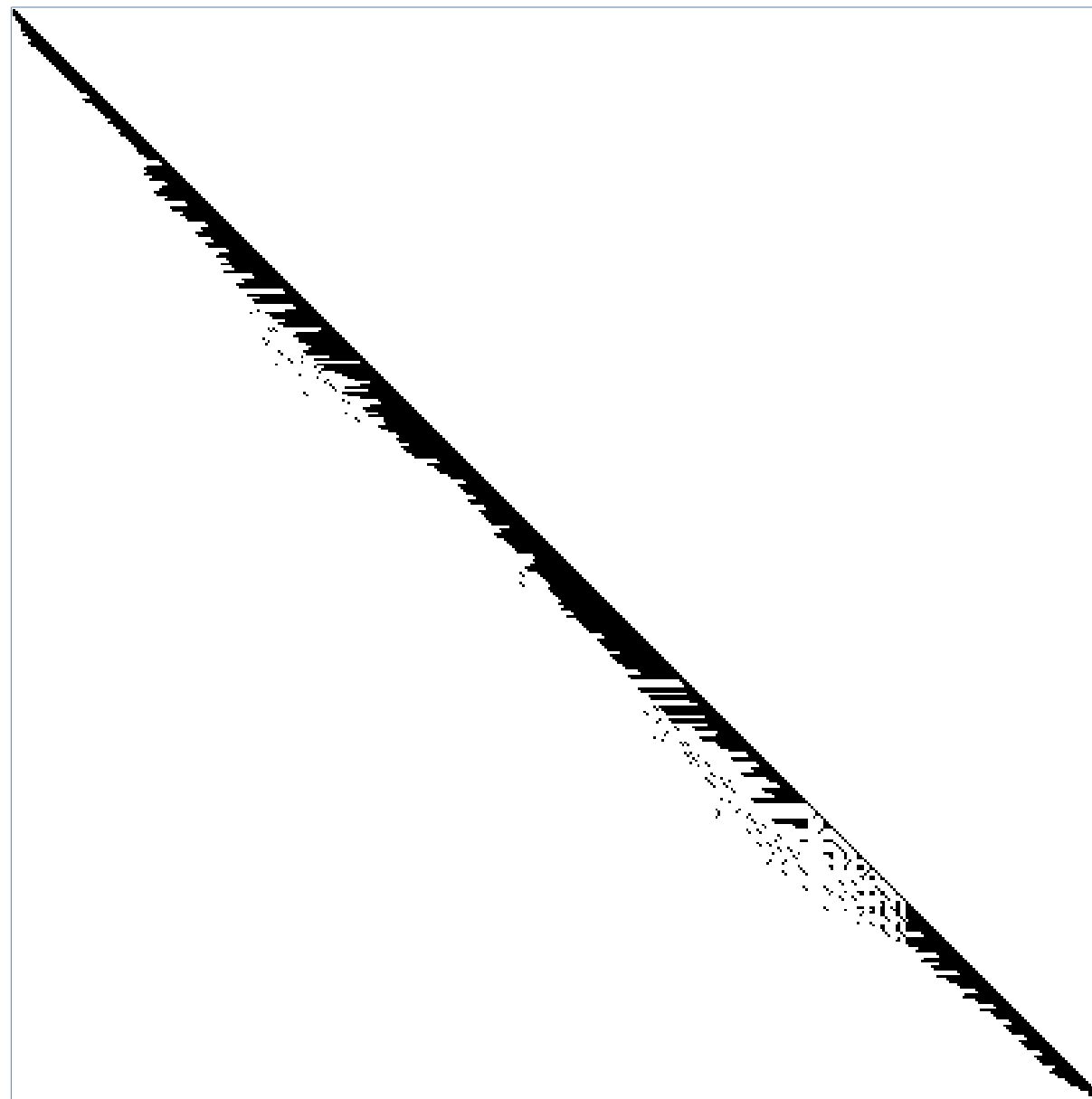
$$\eta(\mathbf{H}_5) = 1767$$

$\mathbf{H}_{10} =$



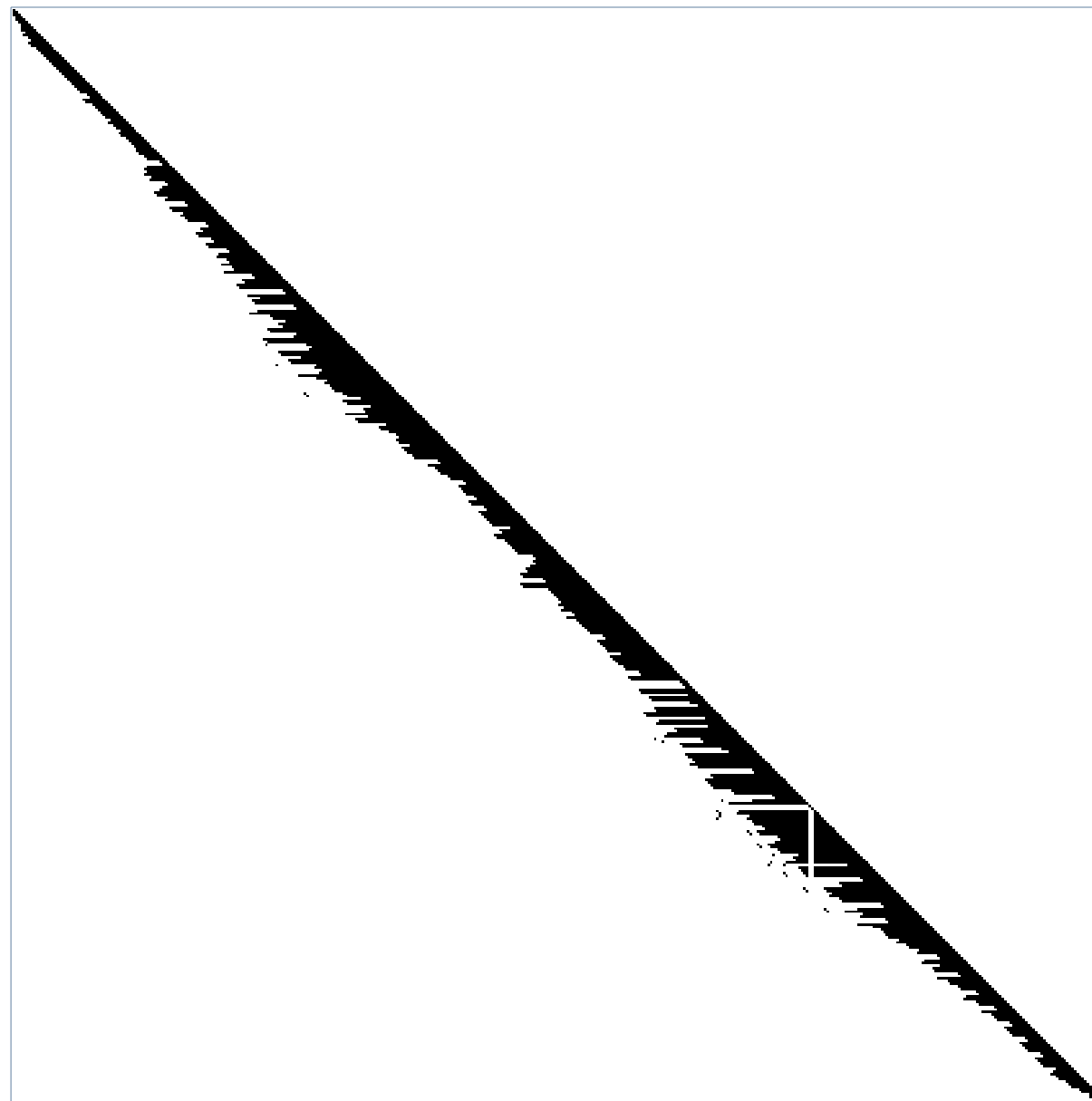
$$\eta(\mathbf{H}_{10}) = 2196$$

$\mathbf{H}_{20} =$



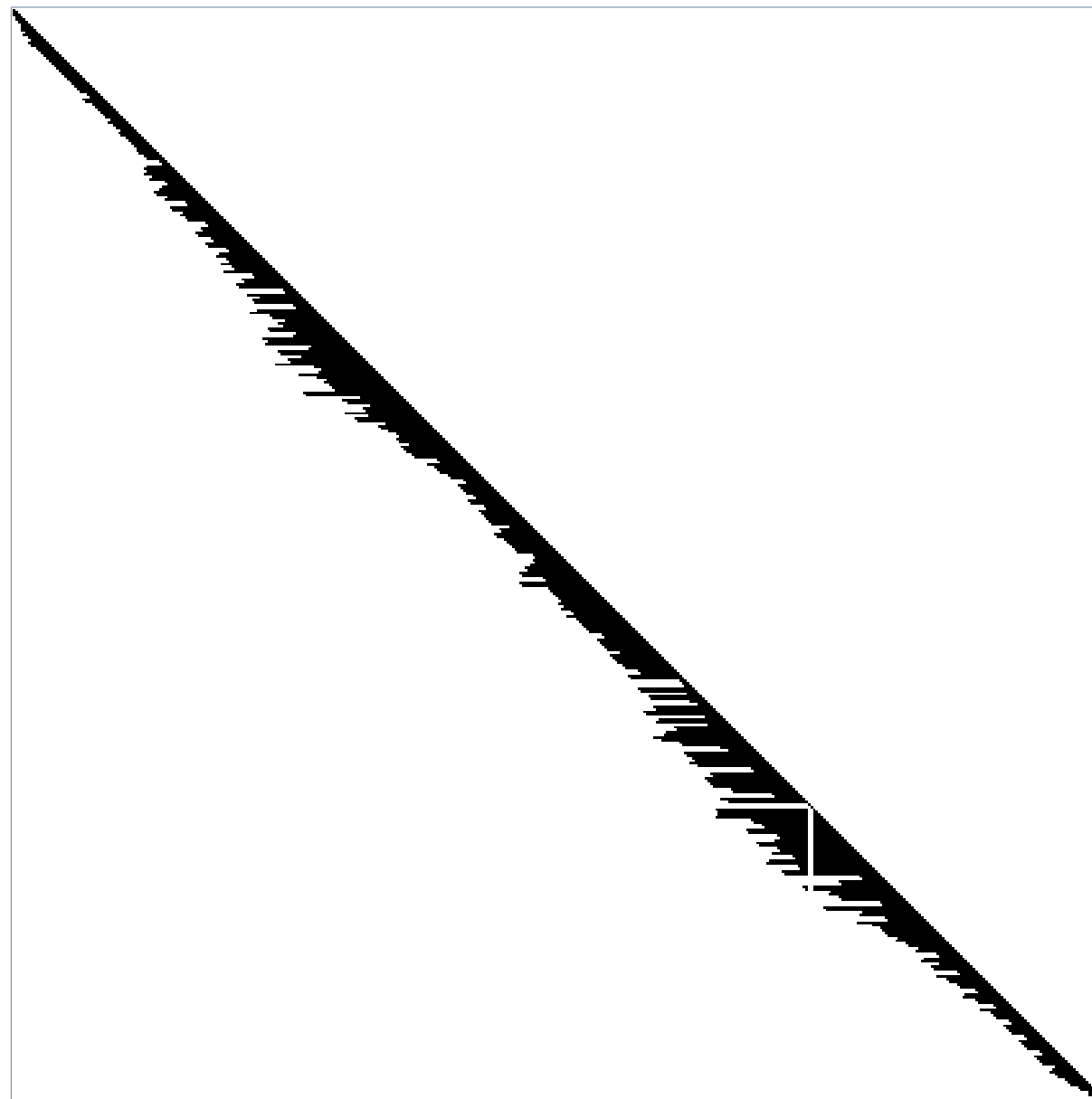
$$\eta(\mathbf{H}_{20}) = 5262$$

$\mathbf{H}_{30} =$



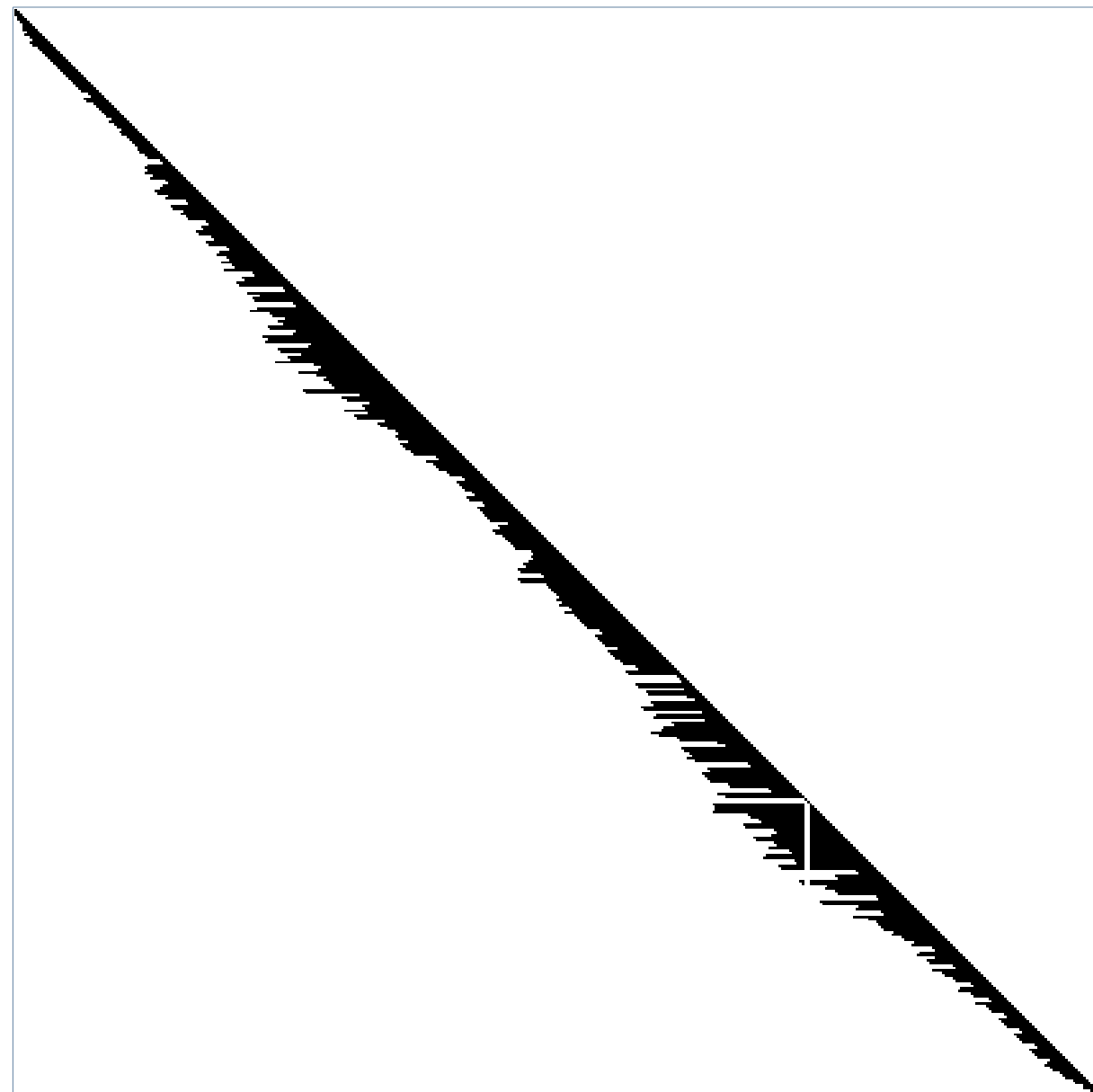
$$\eta(\mathbf{H}_{30}) = 6779$$

$\mathbf{H}_{40} =$



$$\eta(\mathbf{H}_{40}) = 7012$$

$\mathbf{L} =$



$$\eta(\mathbf{L}) = 7012$$

# Factorización Cholesky simbólica truncada

Para una matriz rala  $\mathbf{A}$ , definamos

$$\mathbf{a}_i \stackrel{\text{def}}{=} \{k > i \mid A_{ik} \neq 0\}, i=1 \dots n,$$

como el conjunto de los índices de los elementos no nulos del renglón  $i$  de la parte estrictamente triangular superior de  $\mathbf{A}$ .

De forma análoga definimos para la matriz  $\mathbf{H}_t^T$ , los conjuntos

$$\mathbf{h}_i \stackrel{\text{def}}{=} \{k > i \mid H_{ik} \neq 0\}, i=1 \dots n.$$

Para la matriz ejemplo siguiente, el conjunto  $\mathbf{a}_2$  contendrá  $\{3, 4\}$ ; el conjunto  $\mathbf{h}_2$  contendrá  $\{3, 4, 6\}$ .

$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{12} & & & & A_{16} \\ A_{21} & A_{22} & A_{23} & A_{24} & & \\ & A_{32} & A_{33} & & A_{35} & \\ & A_{42} & & A_{44} & & \\ & & A_{53} & & A_{55} & A_{56} \\ A_{61} & & & & A_{65} & A_{66} \end{pmatrix}$$

$$\mathbf{a}_2 = \{3, 4\}$$

$$\mathbf{H}_t^T = \begin{pmatrix} H_{11} & H_{12} & & & & H_{16} \\ & H_{22} & H_{23} & H_{24} & & H_{26} \\ & & H_{33} & H_{34} & H_{35} & H_{36} \\ & & & H_{44} & H_{45} & H_{46} \\ & & & & H_{55} & H_{56} \\ & & & & & H_{66} \end{pmatrix}$$

$$\mathbf{h}_2 = \{3, 4, 6\}$$

Requeriremos de conjuntos  $r_i$  que serán usados para registrar las columnas de  $\mathbf{H}_t^T$  cuyas estructuras afectarán al renglón  $i$  de  $\mathbf{H}_t^T$ .

```
for  $i \leftarrow 1, 2, \dots, n$ 
·  $r_i \leftarrow \emptyset$ 

for  $i \leftarrow 1, 2, \dots, n$ 
·  $h_i \leftarrow a_i$ 
· for each  $j \in r_i$ 
· ·  $h_i \leftarrow h_i \cup \{x \in h_j \mid x - i < t\} \setminus \{i\}$ 
·  $p \leftarrow \begin{cases} \min\{j \in h_i\} & \text{if } h_i \neq \emptyset \\ i & \text{other case} \end{cases}$ 
·  $r_p \leftarrow r_p \cup \{i\}$ 
```

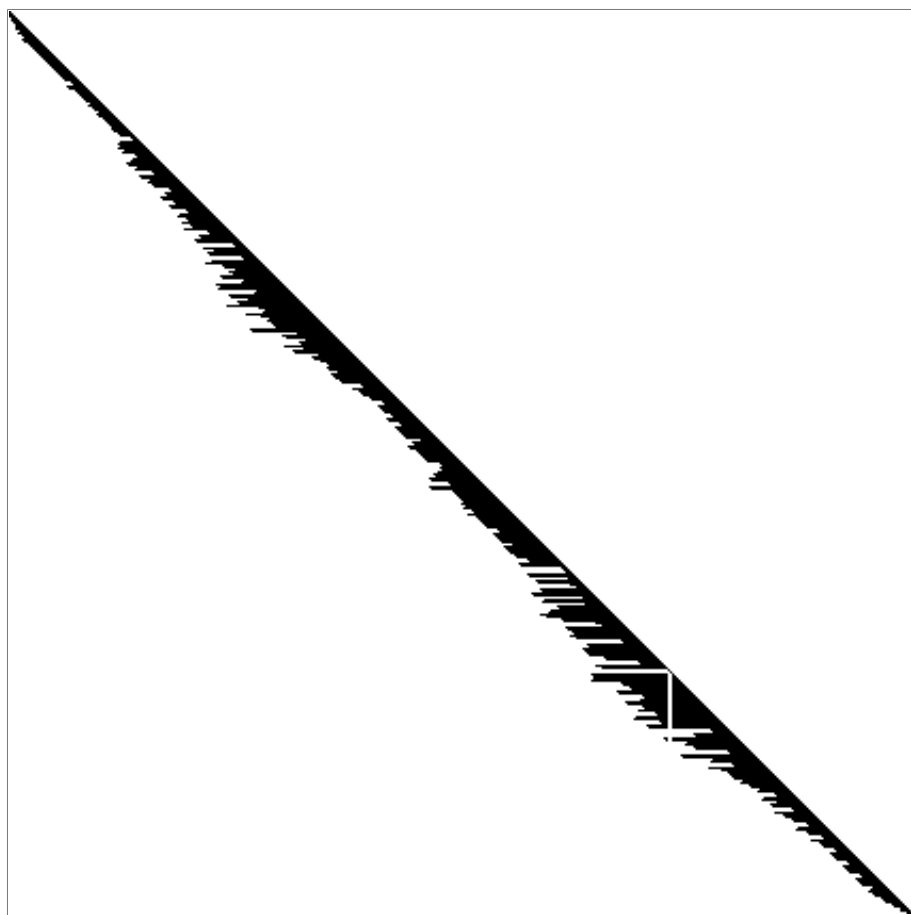
Este algoritmo de factorización simbólica es muy eficiente, la complejidad en tiempo y espacio es de orden  $O(\eta(\mathbf{H}_t))$ .

Una variante más fácil es crear primero la estructura de  $\mathbf{H}_t^T$  y a partir de ésta la de  $\mathbf{H}_t$ .

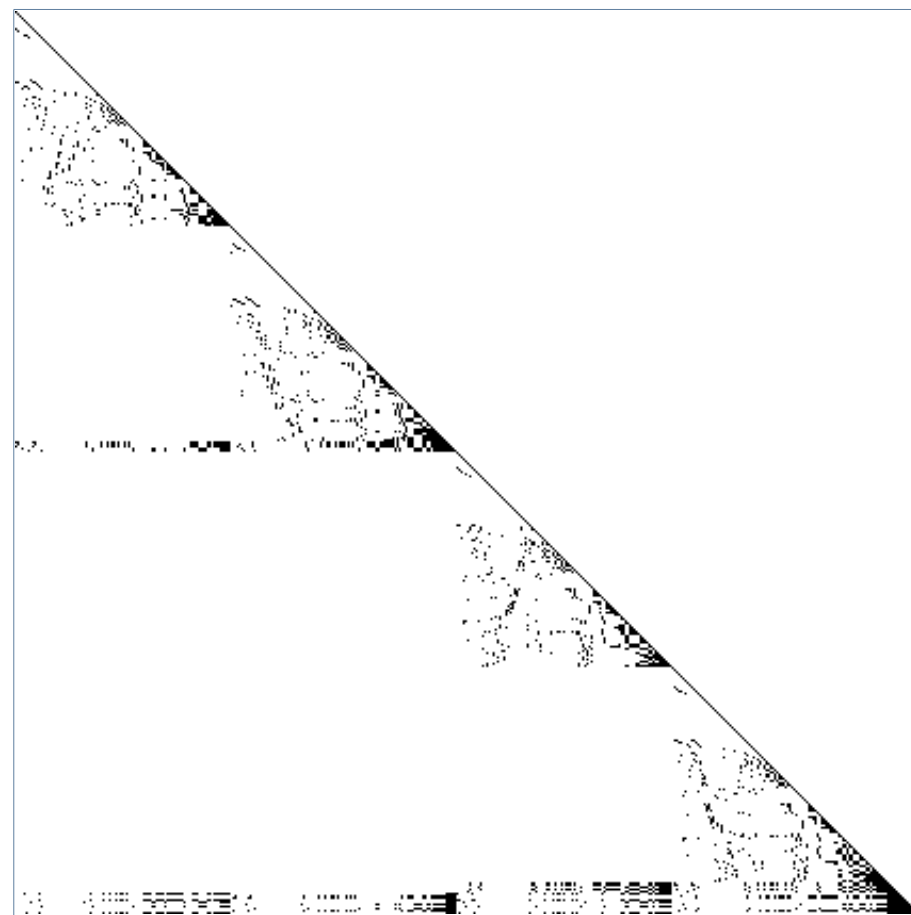


# Reordenamiento

Algunos estudios [Benz02] indican que aplicar reordenamiento al preconditionador deteriora la calidad del preconditionamiento.



$H_n$



$H_n^r$

# El algoritmo sería

El algoritmo del gradiente conjugado con preconditionador Cholesky incompleto sería...

entrada:  $\mathbf{A}$ ,  $\mathbf{x}_0$ ,  $\mathbf{b}$ ,  $\varepsilon$   
 $\mathbf{H}_t, \mathbf{D}, \mathbf{H}_t^T \leftarrow \text{Cholesky}_t(\mathbf{A})$   
 $\mathbf{r}_0 \leftarrow \mathbf{A} \mathbf{x}_0 - \mathbf{b}$   
resolver  $\mathbf{H}_t \mathbf{D} \mathbf{H}_t^T \mathbf{q}_0 = \mathbf{r}_0$   
 $\mathbf{p}_0 \leftarrow -\mathbf{q}_0$   
 $k \leftarrow 0$

mientras  $\|\mathbf{r}_k\| > \varepsilon$

$\mathbf{w} \leftarrow \mathbf{A} \mathbf{p}_k$

$\alpha_k \leftarrow \frac{\mathbf{r}_k^T \mathbf{q}_k}{\mathbf{p}_k^T \mathbf{w}}$

$\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{p}_k$

$\mathbf{r}_{k+1} \leftarrow \mathbf{r}_k + \alpha \mathbf{w}$

resolver  $\mathbf{H}_t \mathbf{D} \mathbf{H}_t^T \mathbf{q}_{k+1} = \mathbf{r}_{k+1}$

$\beta_k \leftarrow \frac{\mathbf{r}_{k+1}^T \mathbf{q}_{k+1}}{\mathbf{r}_k^T \mathbf{q}_k}$

$\mathbf{p}_{k+1} \leftarrow -\mathbf{q}_{k+1} + \beta_{k+1} \mathbf{p}_k$

$k \leftarrow k+1$

# Pero hay un problema...

Puede suceder que el preconditionador no sea SPD.

Para evitar esto utilizaremos el algoritmo de Munksgaard [Munk80], que consiste en dos estrategias:

1. Perturbar la diagonal de  $\mathbf{A}$  con un factor  $\alpha$ ,

$$D_{jj} = \alpha A_{jj} - \sum_{k=1}^{j-1} H_{jk}^2 D_k,$$

esto asegurará que el preconditionador sea SPD, el valor de  $\alpha$  se puede encontrar por prueba y error.

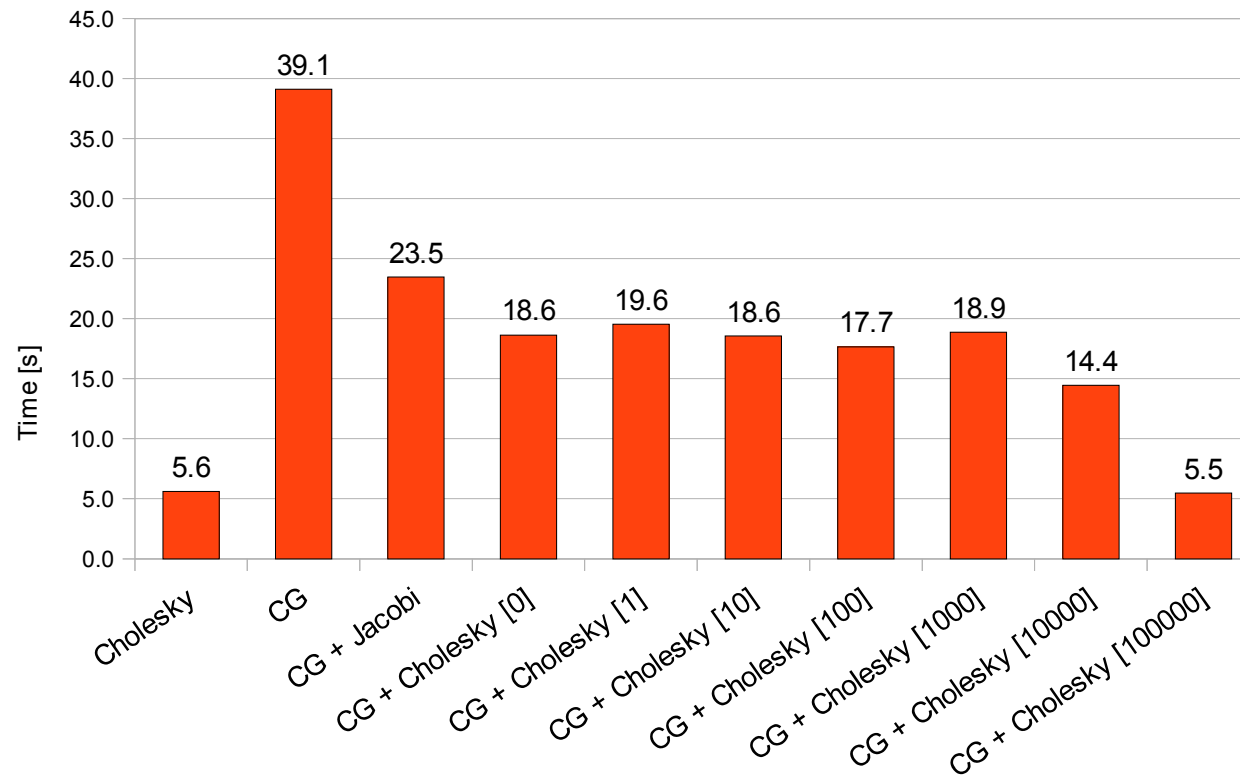
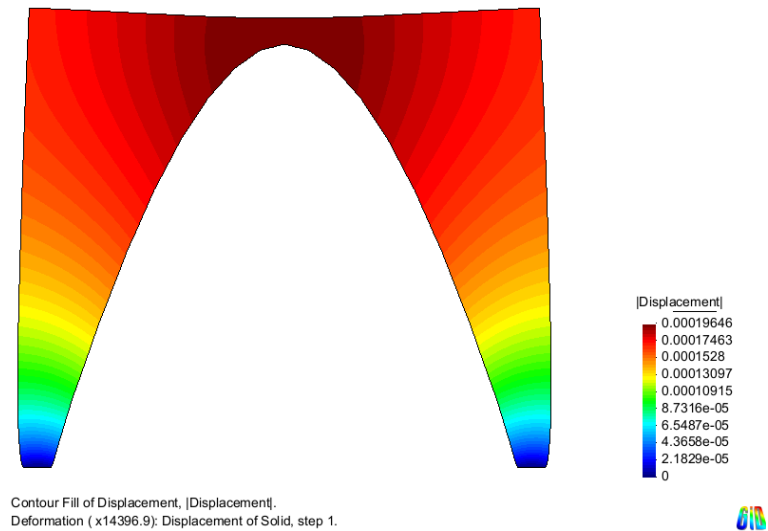
2. Perturbar los pivotes para mejorar la estabilidad si éstos son negativos o cercanos a cero,

$$\text{si } D_{jj} \leq u \left( \sum_{k \neq j} |a_{jk}| \right), \text{ entonces } D_{jj} = \begin{cases} \sum_{k \neq j} |a_{jk}| & \text{si } \sum_{k \neq j} |a_{jk}| \neq 0 \\ 1 & \text{si } \sum_{k \neq j} |a_{jk}| = 0 \end{cases}.$$

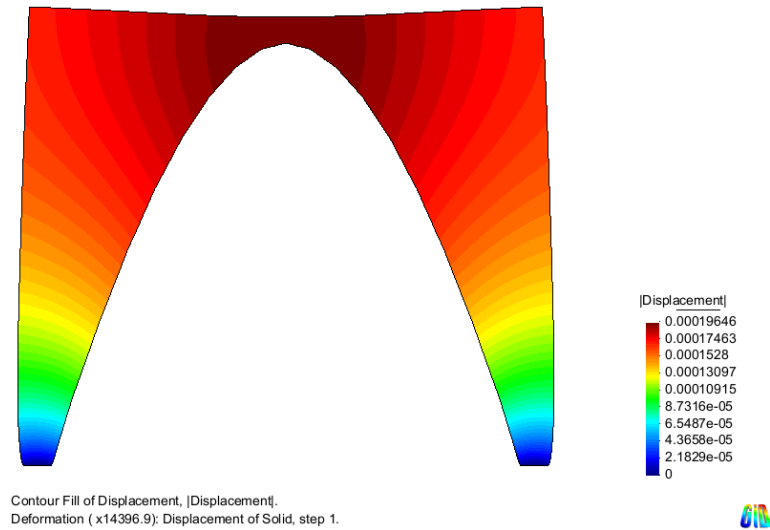
Un valor de  $u$  adecuado es 0.01.

# Algunos resultados

Problem	Arc
Dimension	2
Nodes	50,805
Elements	49,972
Element type	Quadrilateral
Variables	101,610
nnz(A)	1'808,980
Tolerance	$1 \times 10^{-5}$



Problem	Arc
Dimension	2
Nodes	50,805
Elements	49,972
Element type	Quadrilateral
Variables	101,610
nnz(A)	1'808,980
Tolerance	$1 \times 10^{-5}$



Solver	Time [s]	Steps	nnz(H)	Memory	Memory
Cholesky	5.6		6911289	215100454	100.0%
CG	39.1	7723		40768170	19.0%
CG + Jacobi	23.5	4070		40768170	19.0%
CG + Cholesky [0]	18.6	1519	954741	71330422	33.2%
CG + Cholesky [1]	19.6	1519	954741	71330422	33.2%
CG + Cholesky [10]	18.6	1482	990125	72179638	33.6%
CG + Cholesky [100]	17.7	1162	1572311	86152102	40.1%
CG + Cholesky [1000]	18.9	708	3434219	130837894	60.8%
CG + Cholesky [10000]	14.4	308	5715237	185582326	86.3%
CG + Cholesky [100000]	5.5	1	6911289	214287574	99.6%

# ¿Paralelizable?

En cada paso tenemos que resolver el sistema

$$\mathbf{H}_t \mathbf{D} \mathbf{H}_t^T \mathbf{q}_{k+1} = \mathbf{r}_{k+1}.$$

Hagamos  $\mathbf{d} = \mathbf{H}_t^T \mathbf{q}_{k+1}$  y  $\mathbf{c} = \mathbf{D} \mathbf{d}$ , entonces tendremos tres sistemas de ecuaciones

$$\mathbf{H}_t \mathbf{c} = \mathbf{r}_{k+1}, \mathbf{D} \mathbf{d} = \mathbf{c}, \mathbf{H}_t^T \mathbf{q}_{k+1} = \mathbf{d}.$$

Primero tenemos que resolver  $\mathbf{H}_t \mathbf{c} = \mathbf{r}_{k+1}$  para  $\mathbf{c}$  haciendo una sustitución hacia adelante con

$$c_i = \frac{1}{H_{ii}} \left( r_i - \sum_{k=1}^{i-1} H_{ik} c_k \right).$$

Después resolver  $\mathbf{D} \mathbf{d} = \mathbf{c}$ , para obtener  $\mathbf{d}$ .

Finalmente, resolver  $\mathbf{H}_t^T \mathbf{q}_{k+1} = \mathbf{d}$  para  $\mathbf{q}_{k+1}$  sustituyendo hacia atrás con

$$q_i = \frac{1}{H_{ii}^T} \left( d_i - \sum_{k=i+1}^n H_{ik}^T q_k \right).$$

Es muy difícil paralelizar esta sustitución [Heat91 pp454-455] debido a que existe una dependencia secuencial entre las variables.

# ¿Preguntas?

migueltvargas@cimat.mx

# Referencias

- [Benz02] M. Benzi. Preconditioning Techniques for Large Linear Systems: A Survey. *Journal of Computational Physics*, Vol. 182, pp. 418-477. 2002.
- [Golu96] G. H. Golub, C. F. Van Loan. *Matrix Computations*. Third edition. The Johns Hopkins University Press. 1996.
- [Heat91] M T. Heath, E. Ng, B. W. Peyton. Parallel Algorithms for Sparse Linear Systems. *SIAM Review*, Vol. 33, No. 3, pp. 420-460. 1991.
- [Meie94] U. Meier-Yang. Preconditioned conjugate gradient-like methods for nonsymmetric linear systems. University of Illinois. 1992
- [Munk80] N. Munksgaard. Solving sparse symmetric sets of linear equations by preconditioned conjugate gradients. *ACM Transactions on Mathematical Software*, Vol 6-2, pp. 206-219. 1980
- [Noce06] J. Nocedal, S. J. Wright. *Numerical Optimization*, Springer. 2006.
- [Saad03] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.