ALN - Precondicionadores

In. Co.
Facultad de Ingeniería
Universidad de la República

Contenido

- Introducción
- Precondicionamiento
  - Una clasificación
- Precondicionadores Implícitos
  - Jacobi
  - ILU
- Precondicionadores Explícitos
  - Precond. polinomiales
  - Inversa aproximada

Introducción

- Recordando lo visto en clases anteriores, la velocidad de convergencia de los métodos iterativos va a depender fuertemente del número de condición del sistema.

Precondicionamiento

- La idea del precondicionamiento es reducir el número de iteraciones requerido para la convergencia, transformando el sistema original $Ax = b$ en un sistema $A'x = b'$, de tal forma que:
  - Resolver $A'x = b'$ no debe incrementar considerablemente el número de operaciones que se requieren para resolver $Ax = b$.
  - $Ax = b$ y $A'x = b'$ tienen la misma solución, es decir, $A^{-1}b = A'^{-1}b'$.

Precondicionamiento

- La matriz *A'* y el vector *b'* se consiguen premultiplicando o posmultiplicando el sistema por una matriz *M,* llamada *precondicionador* que debe ser fácilmente invertible.

  Por medio del precondicionador se transforma el sistema $Ax = b$ en otro equivalente con condiciones espectrales más favorables reduciendo el número de iteraciones requeridas para la convergencia

Una clasificación

Se pueden distinguir dos grandes grupos de precondicionadores:

- Implícitos
  - Se construye la matriz M y se resuelve Mx=b
    Ej. Jacobi, ILU
- Explícitos
  - Se construye la matriz M^{-1}
    Ej. Polinómicos, inversas aproximadas

Precondicionadores Implícitos

Método de Richardson precondicionado

- Para el sistema precondicionado por la izquierda:

$$M^{-1}Ax = M^{-1}b \quad => \quad Nx = (N - M^{-1}A)x + M^{-1}b$$

El método iterativo general precondicionado quedaría:

$$Nx^k = Nx^{k-1} - M^{-1}(Ax^{k-1} - b)$$

Si tomamos N = I llegamos al método de Richardson precondicionado:

$$x^k = x^{k-1} - M^{-1}(Ax^{k-1} - b)$$

Precondicionador de Jacobi

- Si tomamos M = D = diag(A) y A = L+D+U
  haciendo cuentas llegamos a que:

$$x^k = D^{-1}(-L-U)\, x^{k-1} + D^{-1}b$$

  Este es el método de Jacobi, por esta razón a la matriz M = D se la llama precondiconador diagonal o de Jacobi.

  Se supone que los valores en la diagonal de *A* son diferentes de cero para que M^{-1} pueda estar definida

Precondicionador de Jacobi

- Este método puede ser utilizado sin requerir espacio extra de memoria, es de fácil implementación y fácil de paralelizar, pero hay otros métodos iterativos que logran mejores resultados.

Otros métodos

- Si la matriz A se escribe de la forma $A = D+L+U$, donde $D = diag\ (A)$, L es la parte de A estrictamente triangular inferior y U es la parte de A estrictamente triangular superior, podemos definir dos métodos iterativos precondicionados adicionales:
  - $M = D + L$, en cuyo caso la iteración se llama método de Gauss-Seidel.
  - $M = \omega^{-1}D + LA$, donde ω es un parámetro en el intervalo $(0, 2)$. En este caso la iteración se llama SOR (*successive overrelaxation*).
- Qué pasa con los métodos no estacionarios?

Método del GC precondicionado

Saad deduce el algoritmo de PCG empleando el sistema precondicionado por la Izquierda,y las siguientes ideas:

- El **M**-*producto interior* se define como:

  $(\mathbf{x}, \mathbf{y})_{\mathbf{M}} = (\mathbf{Mx}, \mathbf{y})_2 = (\mathbf{x}, \mathbf{My})_2$, donde $(\cdot, \cdot)_2$ es el usual producto interior euclidiano.

- La matriz $\mathbf{M}^{-1}\mathbf{A}$ es auto-adjunta para el **M**-producto interior:

  $$(\mathbf{M}^{-1}\mathbf{Ax}, \mathbf{y})_{\mathbf{M}} = (\mathbf{Ax}, \mathbf{y})_2 = (\mathbf{x},\mathbf{Ay})_2 = (\mathbf{x},\mathbf{M}^{-1}\mathbf{Ay})_{\mathbf{M}}.$$

Método del GC precondicionado

- Como se vio en clases pasadas la relación

$$\varphi(x) = 1/2\, x^T A x - x^T b$$

alcanza un único mínimo en el vector solución del sistema $Ax = b$

Esta relación se puede reescribir como:

$$\varphi(x) = 1/2\, (r, A^{-1} r)_2$$

Ahora, si reemplazamos en esta igualdad a $r = b - Ax$ por $z = M^{-1} r$, a la matriz A por $M^{-1}A$ y al producto interior euclidiano por el M-producto interior, se obtiene que el método PCG minimiza la función:

$$\varphi(x) = 1/2\, (z, (M^{-1}A)^{-1} z)_M$$

sobre el subespacio de Krylov $\{r0, M^{-1}Ar0, \ldots, (M^{-1}A)k^{-1}r0\}$. Observemos que el precondicionador M necesita ser SDP para que el M-producto interior exista.

Método del GC precondicionado

Aproximación inicial $x_{0.}$ $r_0 = b - Ax_0$

Resolver $Mz_0 = r_{0,}$ $p_0 = z_0$

Mientras $\|r_j\| / \|r_0\| \geq e$ $(j = 0,1,2,3,....)$, *hacer*

$$alfa_j = \frac{\langle r_j, z_j \rangle}{\langle A\, p_j, p_j \rangle}$$

$$x_{j+1} = x_j + alfa_j\, p_j$$

$$r_{j+1} = r_j - alfa_j\, A\, p_j$$

Resolver $Mz_{j+1} = r_{j+1}$

$$beta_j = \frac{\langle r_{j+1}, z_{j+1} \rangle}{\langle r_j, z_j \rangle}$$

$$p_{j+1} = z_{j+1} + beta_j\, p_j$$

fin

Jacobi, GS, SOR y métodos no estacionarios

- Las iteraciones de punto fijo $x^k = x^{k-1} – M^{-1}(Ax^{k-1} - b)$ en resuelven sistemas $M^{-1}Ax = M^{-1}b$.
  - Estos sistemas pueden resolverse también mediante métodos no estacionarios (por ejemplo GC o GMRES) tomando la matriz M^{-1} como precondicionador.
  - M puede ser cualquier matriz no singular pero M^{-1} debe asemejarse a A^{-1}
  - La solución de los sistemas de forma Mx=y no debe ser costosa ya que en estos métodos debe resolverse uno de estos sistemas por cada operación.

Factorizaciones incompletas

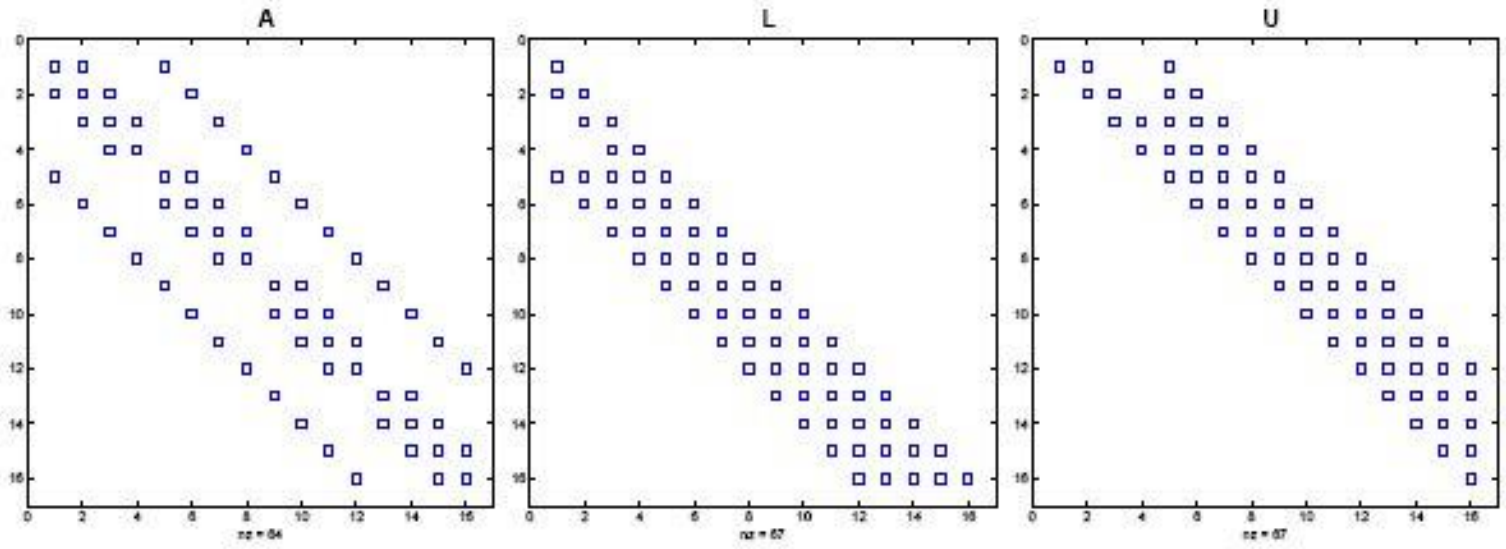
- Una clase importante de precondicionadores se obtiene a partir de la factorización *LU,* que sabemos trabaja para matrices arbitrarias.
  - Sin embargo, existen algunas variantes que explotan la estructura especial que pueda tener la matriz (simétrica, definida positiva, dispersa, etc).

- Para el caso de las matrices dispersas, se tiene un procedimiento denominado factorización incompleta.
  - La idea es obtener matrices M = LU tal que A = LU+R y el sistema Mx=y es fácil de resolver.

Construcción de factorizaciones incompletas

- Si *A* es una matriz dispersa, los factores *L* y *U* usualmente no tienen el mismo patrón de dispersión de la matriz *A* debido al efecto de fill-in visto en el curso.
- Para que el sistema LUx=y sea fácil de resolver, se busca controlar el llenado de los factores L y U.
- La estrategia consiste en descartar los elementos de llenado (**fill-elements**) que aparecen en el proceso de factorización siguiendo cierto criterio.

Factorizaciones incompletas



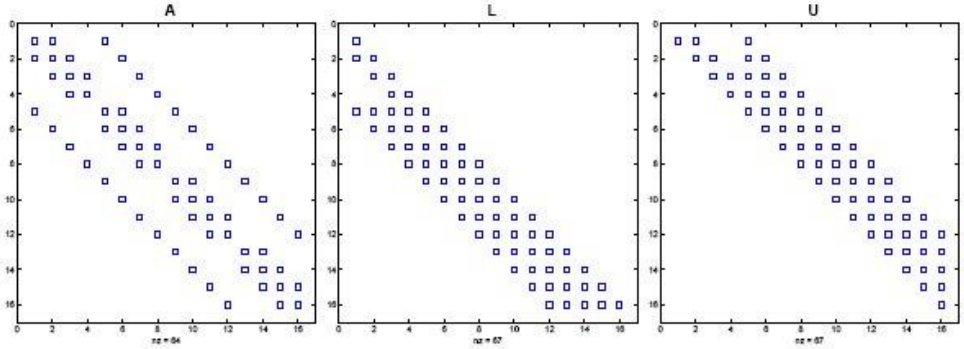
Se puede ver que en L y U existen “diagonales extra” que no están en la parte triangular superior o inferior de la matriz **A**. A tales elementos de estas “diagonales extras” se les denomina elementos de llenado.

- Existen tres formas fundamentales de factorizaciones incompletas:
  - Factorizaciones sin llenado: *ILU*(0)
  - Factorizaciones con llenado, usando como criterio para la introducción del llenado la posición dentro de la matriz: *ILU*(*k)*
  - Factorizaciones con llenado, usando como criterio para la introducción del llenado umbrales numéricos: *ILU*(t)

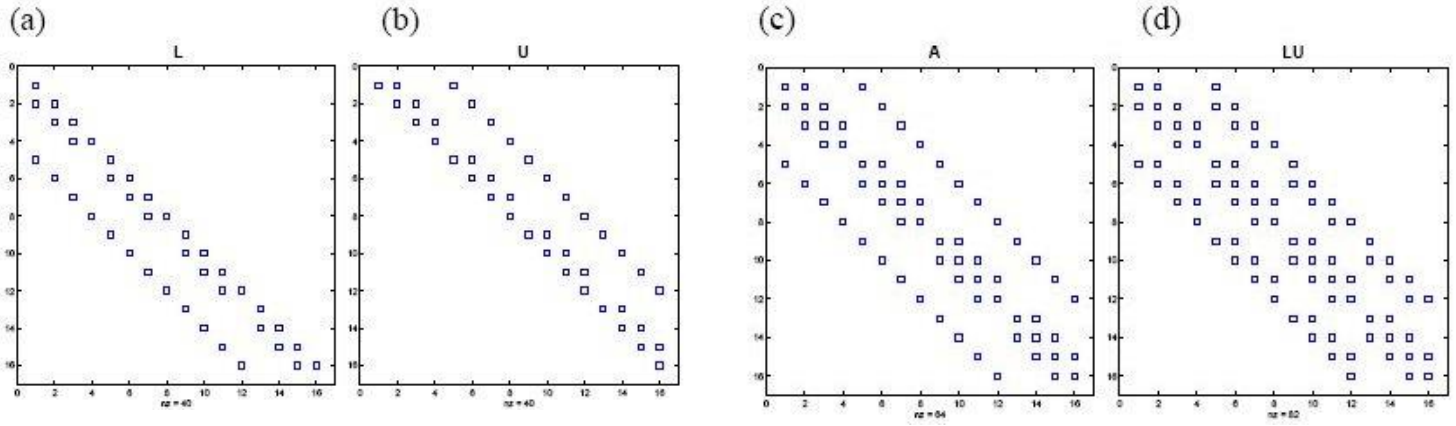
Factorizaciones incompletas ILU(0)

- Es la más sencilla y no introduce llenado alguno, es decir, la factorización incompleta tiene la misma cantidad de elementos no nulos y en las mismas posiciones que en la matriz *A*. Esto significa que en L y U sólo se mantendrán aquellos valores tales que en la misma posición A tiene un valor no nulo.

Factorización completa



Factorización incompleta ILU(0)



En este caso se puede ver que en L y en U se respetan las posiciones de las entradas no nulas de **A**,

Factorizaciones incompletas ILU(k)

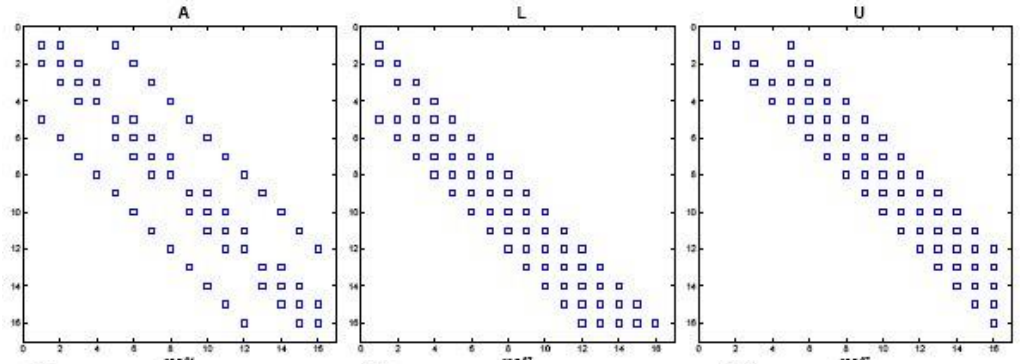
- En estos casos k representa el nivel de llenado que se permite.

  Por ejemplo el nivel-0 es equivalente a la factorización incompleta sin llenado, ILU(0), es decir, el patrón de dispersión de la matriz original.

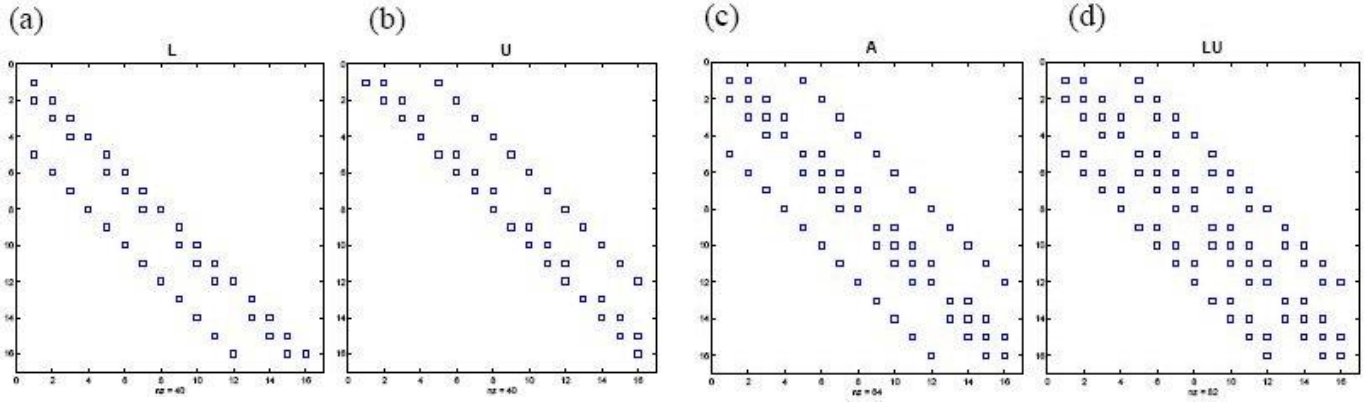
  El *nivel-1 de llenado* incluye al nivel-0 y a los no nulos introducidos por la eliminación de elementos pertenecientes al nivel-0.

  El *nivel-2 de llenado* incluye al nivel-1 y cualquier no nulo producido por la eliminación de los elementos de llenado del nivel-1

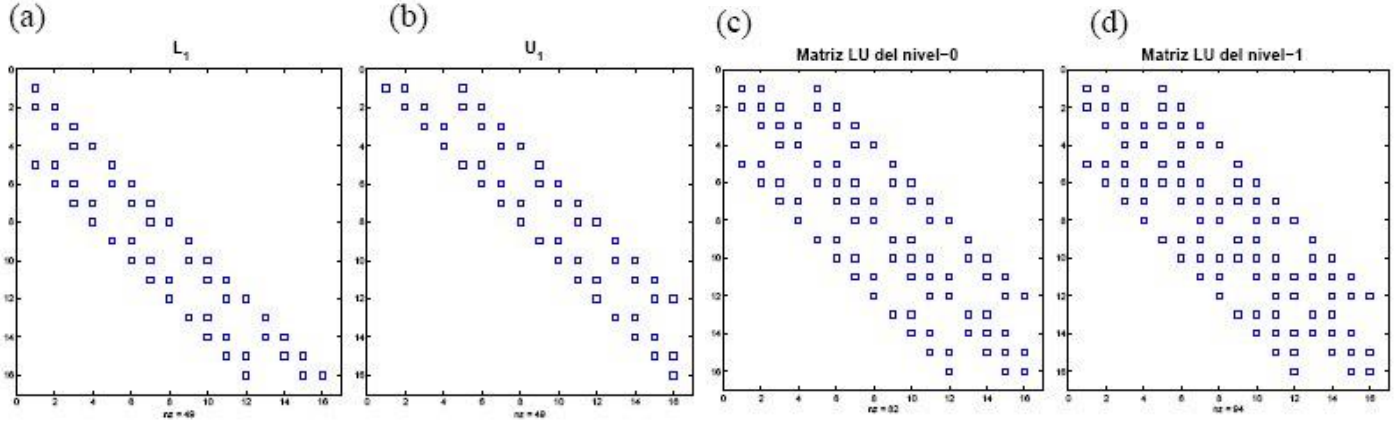
Factorización completa



Factorización incompleta ILU(0)



Factorización incompleta ILU(1)



Se puede ver que por ejemplo las matrices L y U de la factorización ILU(1) respetan la estructura de no nulos de la matriz LU de la factorización ILU(0).

Factorizaciones incompletas ILU(k)

- En general, las factorizaciones de tipo ILU(k) adolecen del defecto de considerar que la importancia numérica de un llenado l_{ij}, depende únicamente de la proximidad topológica entre los nodos i y j, sin tener en cuenta el fenómeno físico que la matriz A representa.

Factorizaciones *ILU(t)*

- Las factorizaciones de este tipo deciden introducir o no un llenado l_{ij} en función de si es superior o inferior a un umbral determinado.
- Dicho umbral se calcula relativo al valor de los elementos de la fila i-ésima de A usando el parámetro *t*.
  - ej: t multiplicado por la norma original de la fila i-ésima.
- Se descartan los elementos de llenado de magnitud inferior al umbral

Factorizaciones *ILU(t)*

- Estas factorizaciones son de aplicación general, pero adolecen de una falta de control fino sobre la cantidad total de llenado que se permite.

- Esta falta de control genera problemas de dimensionamiento del espacio de memoria reservado al precondicionador, y sobre todo, impide un buen compromiso entre complejidad de cálculo del precondicionador y la aceleración que el precondicionador introduce en el método iterativo.

Factorizaciones *ILU(p,t)*

- Variante muy utilizada para mitigar los problemas de la ILU(t).

- Utiliza un criterio de descarte dual:
  - Primero aplica el criterio del umbral
  - Luego se queda con los p elementos de mayor magnitud en la fila que cumplen con el criterio

Precondicionadores Explícitos

Precondicionadores Polinomiales

- Los precondicionadores polinomiales se basan en aproximar la matriz A^{-1} mediante un polinomio de grado m de la matriz A.

- Por ejemplo, si escribimos a como A = I-B y el radio espectral de B es menor que 1, usando las series de Neumann podemos escribir:

$$A^{-1} = \sum_{k=0}^{\infty} B^{k}$$

- Por lo que truncando esta serie se podría obtener una aproximación de la inversa de A.

Precondicionadores Polinomiales

- De forma más abstracta se define un polinomio de grado n de la matriz A: $M^{-1}=P_n(A)$.
- La selección del mejor precondicionador polinómico surge de seleccionar el polinomio que minimiza:

$$\|I - M^{-1} A\|$$

- Por ejemplo, para la norma infinito se utilizan los polinomios de Chebyshev.
  - Requieren estimar límites superiores e inferiores del espectro de A, lo cual no siempre es fácil.

Precondicionadores Polinomiales

- Si bien la construcción de este tipo de precondiccionadores únicamente requieren productos matriz-vector con A y tienen un excelente potencial de paralelización no son tan efectivos en la reducción del número de iteraciones como los métodos de factorización incompleta

- Con los precondicionadores polinomiales, la reducción del número de iteraciones tiende a compensarse con el costo de los productos matriz-vector que se efectúan en cada iteración.

Precondicionadores Polinomiales

- Con más precisión, Axelsson demostró que el costo por iteración aumenta linealmente con el número k + 1 de términos en el polinomio, mientras que el número de iteraciones disminuye más lentamente que O (1/ k + 1).

- Por lo tanto, los precondicionadores polinomiales no pueden ser muy efectivos, especialmente en ordenadores en serie o de memoria compartida.

- Para ordenadores en paralelo con memoria distribuida, estos precondicionadores son un poco más atractivos debido al número reducido de productos escalares.

Inversas aproximadas

- La idea básica de estos precondicionadores es buscar una matriz M tal que AM sea tan cercana a la identidad como sea posible.

  Una idea de cómo se plantea este problema es la siguiente:

Inversas aproximadas

- Se construye una matriz inversa aproximada usando el producto escalar de Frobenius.

  Sea $S \subset M_n$, el subespacio de las matrices M donde se busca una inversa aproximada explícita. La formulación del problema es:

  encontrar $M_0 \in S$ tal que:

$$M_0 = \min_{M \in S} \| AM - I \|_F \quad (1)$$

  donde **I** es la matriz identidad, **A** es una matriz no simétrica y F denota la norma de Frobenius.

Inversas aproximadas

- La norma de Frobenius está definida por

  $\|\mathbf{A}\|_F^2 = \langle \mathbf{A}, \mathbf{A} \rangle_F$ con $\langle \mathbf{A}, \mathbf{B} \rangle_F = \mathrm{tr}(\mathbf{A}\mathbf{B}^T)$.

  Esta norma tiene la virtud de satisfacer la propiedad

  $$\| \mathbf{AM-I} \|_F^2 = \sum_{k=1}^{N} \|\mathbf{A}\mathbf{m}_k - \mathbf{e}_k\|_2^2 \quad (2)$$

  con $\mathbf{m}_k$ y $\mathbf{e}_k$ las k-ésimas columnas de **M** y **I**, respectivamente.
- Así, basta con resolver en forma independiente N problemas de mínimos cuadrados para encontrar la solución de (1). De esa manera, las columnas de la inversa aproximada **M** pueden calcularse y aplicarse en paralelo.

Inversas aproximadas

- Para resolver (2) de forma eficiente es crucial la observación de que la inversa de A usualmente es mucho más densa que A, pero los coeficientes significativos de A^{-1} son únicamente unos pocos. Es decir, se espera que M sea una matriz dispersa. Por lo tanto el problema (2) es en realidad de una dimensión muy pequeña y puede ser resuelto en forma eficiente.