Incomplete factorization preconditioners and their updates with applications - II[1,2]

Daniele Bertaccini, Fabio Durastante

Moscow – August 25, 2016

Università di Roma Tor Vergata

Università degli Studi dell'Insubria
Department of Science and High Technology

Notes of the course: *"Incomplete factorization preconditioners and their updates with applications"*.
Lesson 2. Inverting Sparse Matrices, Decaying Pattern, Inverse LU factorization

> An "A^{-1}" in a formula almost always means "solve a linear system" and almost never means "compute A^{-1}."
>
> Golub–Van Loan

In this lecture we are going to treat the topic of **inverse approximation of the matrix** A, namely the calculation of a preconditioner based on computing efficiently a sparse approximation of A^{-1}. Differently from what we have done in the precedent lecture we are going to treat the so-called *explicit* preconditioning technique, that is explicit because it does not request the solution of a sparse triangular system as in the *implicit* case[3]. It relies only upon **sparse matrix-vector product**. As usual, this approach have some improving and some drawbacks. Start observing that having not to solve for triangular sparse system we can not encounter serial bottleneck, so, in the phase of implementation, parallel architecture can be viable[4]. Another advantage could arise in the case of poorly conditioned residual matrix in the incomplete factorization, or if A is far from a diagonally dominant matrix. In this cases the implicit preconditioning with $A = \tilde{L}\tilde{U} - R$ give rise to the calculation of:

$$\tilde{L}^{-1} A \tilde{U}^{-1} = I - \tilde{L}^{-1} R \tilde{U}^{-1},$$

and the matrix $L^{-1}RU^{-1}$ could have eigenvalue far from being clustered around zero. To account for this problem *explicit* preconditioner are come into attention. Nevertheless this strategy is far from being trouble-free, ascertain that a sparse inverse of A is not singular is a topic to be accounted for, also treating with non symmetric problem could be a not so easy task. Preconditioner generated for being applied on the left can be ineffective on the right and vice–versa. Another problem can arise for A with many entries of great magnitude, attempt calculating a sparse approximation of A^{-1} could lead to a matrix with small entries and poorly effective on the problem. At last, we have to observe that, *assuming the use of a standard computing architecture*, the time needed for computing an **explicit** preconditioner is usually greater than the one needed for computing an *implicit* one[5].

There are many techniques for obtaining preconditioners in this form, for example, we can remember[6]

[3] The preconditioner of the previous lecture are implicit preconditioner, because at each step of the preconditioned algorithm we need to solve for auxiliary linear systems with matrix P.

[4] Daniele Bertaccini and Salvatore Filippone. Sparse approximate inverse preconditioners on high performance gpu platforms. *Computers & Mathematics with Applications*, 71(3):693 – 711, 2016. ISSN 0898-1221. .

[5] While using parallel implementation on GPU architecture like in (Bertaccini and Filippone, 2016) can boost the performance.

[6] For a comparative study on this various technique you can look at (Benzi and Tuma, 1999) or (Bertaccini and Durastante, 2017).

Frobenius norm minimization techniques that is the computation of the preconditioner M^{-1} as the one that satisfies

$$\begin{aligned} M^{-1} &= \arg\min_{G\in\mathcal{S}} F(G) = \arg\min_{G\in\mathcal{S}} \|I - AG\|_F^2 = \\ &= \arg\min_{G\in\mathcal{S}} \sum_{i=1}^{n} \|\mathbf{e}_i - A\mathbf{g}_i\|_2^2, \end{aligned} \tag{1}$$

over a certain set $\mathcal{S}$ of sparse matrices of given pattern,

Neumann series type preconditioner, in which the preconditioner M^{-1} is expressed as a particular polynomial

$$M^{-1} = p_k(A),\ p_k(z) \in \mathbb{P}_{\leq k}[x], \tag{2}$$

satisfying some requisites on the spectrum of the A matrix,

Sparse inversion of sparse triangular factor in which the objective is performing a sparse inversion technique on the triangular factors of an implicit preconditioner.

Incomplete biconjugation methods that is an approach built upon a direct approximate factorization of the matrix A^{-1}.

In the following we are going to focus on the *sparse inversion of sparse triangular factor*, while the next lesson will be focused on exploiting *incomplete biconjugation methods*.

Before the presentation of the algorithm we will need to do some preliminary work and observations. So, let us start with answering the question of why we need to pre-pose the word *sparse* to the word *inverse*. We are working with matrices A that are *sparse*, therefore we want, both from the point of view of memory occupation and computational complexity, to work with preconditioner M^{-1} that are still sparse. Nevertheless, the inverse of sparse matrix A can be no more sparse, i.e., can become dense also for matrices A with very few elements.

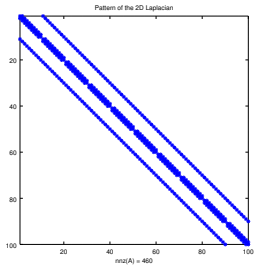
Therefore, the next point of our discussion, will be to characterize this behaviour in terms of property of the matrix A, to see if there is any chance of obtaining a *sparse inverse* of the matrix of our linear system.

To clarify this statement we will follow the approach given in (Gilbert, 1994) in which the language of graph theory is used. In a certain sense this is the natural language in which this kind of structural results have to be treated.

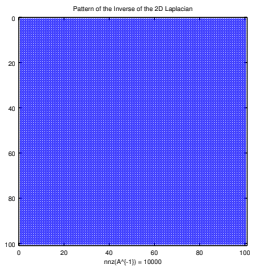
Starting from a sparse matrix $A \in \mathbb{R}^{n\times n}$ we will consider a particular graph:

Definition 1: (struct(A))

Given a sparse matrix $A \subset \mathbb{R}^{n\times n}$ we consider the graph $G(A)$, called the structure of A, or $G(A) = \text{struct}(A)$ defined by the

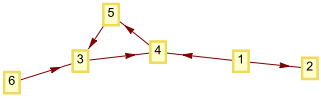


(a) Pattern of the 2D Laplacian



(b) Pattern of the Inverse of the 2D Laplacian

Figure 1: A case in which the inverse of a sparse matrix A, i.e., the five points discretization of the laplacian over a square, is a full matrix.



(a) Example of a graph G

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

(b) Adjacency matrix of G

Figure 2: Example of a generic graph with its adjacency matrix.

vertex set V and edge set E:

$$V = \{i : 1 = 1, \ldots, n\},$$
$$E = \{(i,j) : i \neq j \text{ and } A_{i,j} \neq 0\}.$$

Definition 2: ($struct(\mathbf{x})$)

Given a vector $\mathbf{x} \in \mathbb{R}^n$ the structure of the vector, in respect to a the structure of the matrix $A \subset \mathbb{R}^{n\times n}$ represented by the graph $G(A)$, is the set:

$$\text{struct}\,(\mathbf{x}) = \{i : x_i \neq 0\},$$

that is interpreted as a subset of the set of vertices in $G(A)$.

And now we need some other definition related to graph theory[7].:

Definition 3

We say that the structure struct(x) of the vector $\mathbf{x} \in \mathbb{R}^n$ is **closed** in respect to the matrix graph $G(A)$ if there is no edge of $G(A)$ from a vertex not in struct$(\mathbf{x})$ to a vertex in struct$(\mathbf{x})$, i.e., if and only if $x_j \neq 0$ and $(A)_{i,j} \neq 0$ implies $x_i \neq 0$. Therefore we can define the **closure** of struct$(\mathbf{x})$ in $G(A)$ as

$$\text{closure}(\mathbf{x}) = \cap \{\mathbf{y} : \text{struct}(\mathbf{x}) \subseteq \text{struct}(\mathbf{y}) \text{ and } \mathbf{y} \text{ is closed}\}.$$

At last the **transitive closure** of A is the graph $G^*(A)$ with edges corresponding to paths in $G(A)$, that is:

$$i \overset{G^*(A)}{\longrightarrow} j \;\Leftrightarrow\; i \neq j \text{ and } i \overset{A}{\Longrightarrow} j.$$

In the last we says that G is **strongly connected** if its closure is a complete directed graph, that is $\forall i, j \in V_{\text{closure}(G)}$ is such that $i \overset{\text{closure}(G)}{\Longrightarrow} j$. The matrix A is **irreducible** if $G(A)$ is strongly connected.

Definition 4

A finite set of complex numbers $\{\xi_1, \ldots, \xi_n\}$ is **algebraically independent** if the the point $(\xi_1, \ldots, \xi_n)$ is not a zero of any non–zero polynomial of n variables with integer coefficients, that is:

$$\{\xi_1, \ldots, \xi_n\} : \forall\, p \in \mathbb{Z}[x_1, \ldots, x_n] \text{ holds } p(\xi_1, \ldots, \xi_n) \neq 0,\ p \not\equiv 0.$$

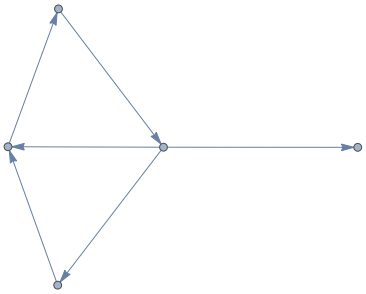
Theorem 1

Let the structure of A and $\mathbf{b}$ be given. Then we have that:

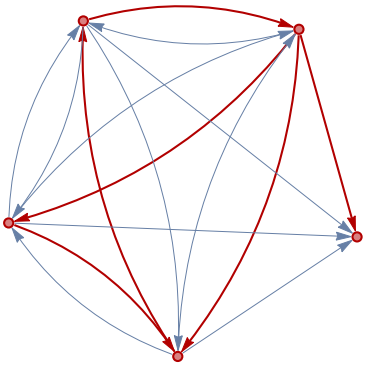
1. Irrespective of the values of the nonzeros in A and in $\mathbf{b}$, if A

[7] The following notation is established:

- Given two graph $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ we have that $G_1 \subseteq G_2 \Leftrightarrow V_1 \subseteq V_2$ and $E_1 \subseteq E_2$,
- $i \overset{A}{\longrightarrow} j$ there exists an arc from i to j in $G(A) \Leftrightarrow (A)_{i,j} = a_{i,j} \neq 0$,
- $i \overset{A}{\Longrightarrow} j$ there exists a path from i to j in $G(A)$, a consecutive set of vertexes linking i to j, it may have length zero (case $i = j$). $\Leftrightarrow (A)_{i,j} = a_{i,j} \neq 0$.



(a) Graph $G = (E, V)$



(b) Transitive Closure of G

Figure 3: Example of the transitive closure (completed in blue) of the graph G (highlighted in red).

is nonsingular then: $\text{struct}(A^{-1}\mathbf{b}) \subseteq \text{closure}(\mathbf{b})$.

2. There exist nonzero values for which the above inclusion is actually an equality.

Proof

We will prove the two statements in sequence:

1. Consider nonzero values such that A is nonsingular and the apply a renumbering of the elements such that $\text{closure}(\mathbf{b}) = \{1, 2, \ldots, k\}$ for some $k \leq n$, now we can rewrite $A\mathbf{x} = \mathbf{b}$ as:

$$\begin{pmatrix} B_{k\times k} & D_{k\times(n-k)} \\ C_{(n-k)\times k} & E_{(n-k)\times(n-k)} \end{pmatrix} \begin{pmatrix} \mathbf{y}_k \\ \mathbf{z}_{n-k} \end{pmatrix} = \begin{pmatrix} \mathbf{d}_k \\ \mathbf{0}_{n-k} \end{pmatrix}.$$

   By the definition of closure (3) $\nexists\, e = (i, j) \in E$ with $i \notin \text{closure}(\mathbf{b})$ and $j \notin \text{closure}(\mathbf{b}) \Rightarrow C = 0$.

   Now A is non singular $\Rightarrow$ E is non singular $\Rightarrow$ $\mathbf{z} = \mathbf{0} \Rightarrow$ $\text{struct}(\mathbf{x}) \subseteq \{1, 2, \ldots, k\} = \text{closure}(\mathbf{b})$.

2. Now we choose a set of algebraically independent values for the nonzeros of A, this implies that A is non singular. Then we choose also $b_i = 1$ if $i \in \text{struct}(\mathbf{b})$. Let $\mathbf{x} = A^{-1}\mathbf{b}$ and, similarly to what we have done in the precedent step, renumber A so that $\text{struct}(\mathbf{x}) = \{1, 2, \ldots, k\}$ for some $k \leq n$ and rewrite $A\mathbf{x} = \mathbf{b}$ as:

$$\begin{pmatrix} B_{k\times k} & D_{k\times(n-k)} \\ C_{(n-k)\times k} & E_{(n-k)\times(n-k)} \end{pmatrix} \begin{pmatrix} \mathbf{y}_k \\ \mathbf{0}_{n-k} \end{pmatrix} = \begin{pmatrix} \mathbf{d}_k \\ \mathbf{e}_{n-k} \end{pmatrix},$$

   Rewriting the matrix-vector product for C we have that:

$$\sum_{1\leq j\leq k} c_{i,j} y_j = e_i.$$

   Matrix B is nonsingular because of the choice of the set of algebraically independent values for the nonzeros of A done previously; then we have that $B\mathbf{y} = \mathbf{d}$ implies, by the *Cramer Rule*, that $y_j = \det(B|_j^{\mathbf{d}})/\det(B)$ and then:

$$\sum_{1\leq j\leq k} c_{i,j} \det(B|_j^{\mathbf{d}}) - e_i \det(B) = 0.$$

   $A_{n\times n}\mathbf{x} = \mathbf{b} \Rightarrow x_i = \frac{\det(A|_i^{\mathbf{b}})}{\det(B)}$ for $i = 1, \ldots, n$ where $A|_i^{\mathbf{b}}$ is the matrix A with the i-column replaced by $\mathbf{b}$.

   Now this is a polynomial with rational coefficients in the entries of A matching zero, so it has to be the zero polynomial, but $y_j \neq 0\ \forall\, j = 1, \ldots, k \Rightarrow \det(B|_j^{\mathbf{d}}) \neq 0$ as a polynomial, therefore we have $c_{i,j} = 0 \Rightarrow C = 0$. So $\mathbf{x}$, partitioned as above, is closed, besides $\det(B) \neq 0$ and the $e_i = \mathbf{0}$. Iterating the argument for all i we have $\mathbf{e} = \mathbf{0}$, that is:

$$\mathbf{b} = \begin{pmatrix} \mathbf{d} \\ \mathbf{0} \end{pmatrix},$$

and:

$$\text{struct}(\mathbf{b}) \subseteq \text{struct}(\mathbf{x}) = \text{closure}(\mathbf{x}) \Rightarrow \text{closure}(\mathbf{b}) \subseteq \text{closure}(\mathbf{x})$$

together with the first part of the theorem, this proves $\text{closure}(\mathbf{b}) = \text{struct}(\mathbf{x})$.

Now, as a corollary of the previous theorem we can state the result for an irreducible nonsingular sparse matrix. Considering that the j-th column of the graph $G^*(A)$[8] is the closure of j-th vector of the canonical base, namely $\text{closure}(\mathbf{e}_j)$ we have:

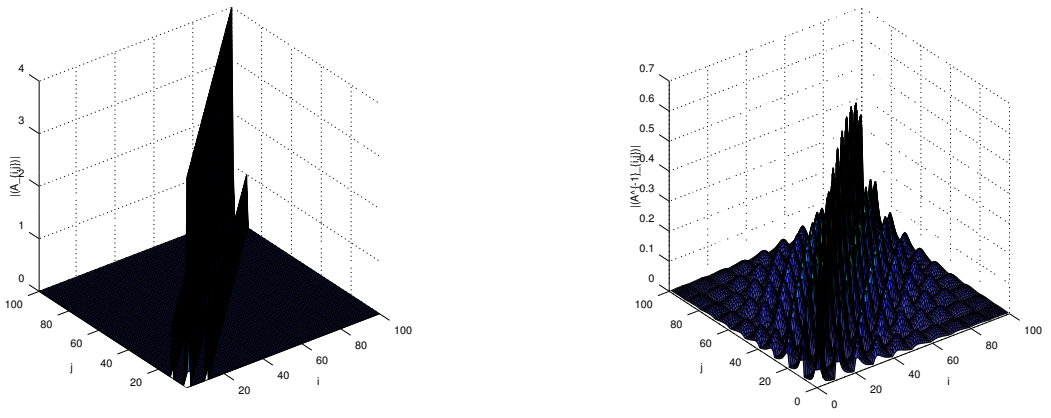
[8] $G^*(A)$ is the *transitive closure* of A see (3).

Theorem 2

Given the structure of a sparse matrix A, $G(A) = \text{struct}(A)$ we have:

1. Irrespective of the values of the nonzeros in A, if A is nonsingular then $G(A^{-1}) \subseteq G^*(A)$.
2. There exist nonzero values for which the above inclusion is actually an equality.

So we have reasonably no expectation for the sparsity of the inverse matrix. Nevertheless the information we have obtained for the matrix A^{-1} involves only the position of the non-zero elements. We don't know anything about the magnitude of such elements. Let us look at figure 4. We have plotted the absolute value of the elements of the inverse of the matrix A being the five points discrete 2D laplacian.



(a) Cityplot of the 2D Laplacian

(b) Cityplot of the Inverse of the 2D Laplacian

Figure 4: Decay of the element of the inverse of a banded matrix. The case of the discrete laplacian.

As we observe the elements of the inverse matrix are all different from zero, but their magnitude shows a decay along the diagonals starting from the main. As a next step we are going to look into this behaviour with a greater detail.

Bounds for the elements of A^{-1}

Of the many possible cases for which this kind of results exists we are going to focus on the case of A banded, starting with A being also and positive definite, for which we can apply the results in (Demko et al., 1984)[9]:

[9] Further generalization of this results are in (Eijkhout and Polman, 1988) and (Nabben, 1999). Other extensions, tanking also into account the possibility of having matrices with more structure, are in (Canuto et al., 2014). We focus on the banded case because it brings fort some of the fundamental techniques for the more general cases.

Theorem 3

Let A and A^{-1} be in $\mathcal{B}(l^2(s))$. Then if A is positive definite and m-banded we have that:

$$(|A^{-1}|)_{i,j=1}^{n} = |a_{i,j}^{-1}| \leq C\lambda^{|i-j|},$$

where:

$$\lambda = \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^{2/m},$$

and:

$$C = \|A^{-1}\| \max \left\{ 1, \frac{\left(1 + \sqrt{\kappa(A)}\right)^2}{2\kappa(A)} \right\}$$

If A fails to be positive definite but is still m-banded, quasi-centered, bounded, and boundedly invertible then:

$$(|A^{-1}|)_{i,j=1}^{n} = |a_{i,j}^{-1}| \leq C_1 \lambda_1^{|i-j|},$$

where

$$\lambda_1 = \left(\frac{\kappa(A) - 1}{\kappa(A) + 1} \right)^{\frac{1}{m}},$$

and

$$C_1 = (m+1)\lambda_1^{-m} \|A^{-1}\| \kappa(A) \max \left\{ 1, \frac{1}{2} \left[\frac{1 + \kappa(A)}{\kappa(A)} \right]^2 \right\}.$$

To prove this statement we need some preliminary work. In total generality we can start with a general complex, separable, Hilbert space H, and let $\mathcal{B}(H)$ denote the Banach algebra of all linear operators on H that are also bounded. Now if $A \in \mathcal{B}(H)$ then we can represent A as a matrix with respect to any complete orthonormal set. In this way, having chose a representation, we may regard A as an element of $B(l^2(S))$, where $S = \{1, 2, \ldots, N\}$. In this space the usual matrix product define the action A over the space. Now we can intend A as a matrix representing a bounded operator in $\mathcal{B}(l^2(S))$. For such matrices A we will say that A is *m-banded* if there is an index l such that:

$$a_{i,j} = 0, \quad \text{if } j \notin [i - l, i - l + m].$$

We will say that A is *centered* and *m-banded* if m is even and the l above may be chosen to be $m/2$. In this case we have that that the zero elements of the *centered* and *m-banded*[10] are:

[10] A selfadjoint matrices are naturally centered, i.e. a tridiagonal selfadjoint matrix is centered and 2-banded.

$$a_{i,j} = 0, \quad \text{if } |i-j| > \frac{m}{2}.$$

Now let $\mathbb{P}_{\leq n}$ denote, as usual, the polynomial of degree less than or equal to n. If $K \subseteq \mathbb{C}$ and f is a fixed complex-valued function on K we define the norm:

$$\|f\|_K = \sup_{z \in K} |f(z)|$$

and the relative approximation error for the set of polynomial $\mathbb{P}_{\leq n}$ to an f over the set K as:

$$e_n(K) = \inf_{p \in \mathbb{P}_{\leq n}} \|f - p\|_K.$$

To proceed we need a results due to Chebyshev, (Tchebychev, 1907), and Bernstein, (Bernstein, 1926), for which a modern presentation is in (Meinardus and Schumaker, 1967):

Theorem 4

Let $f(x) = 1/x$ and let $0 < a < b$. Set $r = b/a$ and:

$$q = q(r) = \frac{\sqrt{r}-1}{\sqrt{r}+1}$$

then:

$$e_n([a,b]) = \frac{(1+\sqrt{r})^2}{2ar} q^{n+1}$$

And with this we can prove the results needed to have our proof.

Theorem 5

Let A be a positive definite, m-banded, bounded and boundedly invertible matrix in $l^2(S)$. Let $[a,b]$ be the smallest interval containing $\sigma(A)$. Setting $r = b/a$, $q = q(r)$ as in Theorem (4), and set $C_0 = (1+\sqrt{r})^2/(2ar)$ and $\lambda = q^{2/m}$. Then we have:

$$|A^{-1}| = (|a^{-1}_{i,j}|)_{i,j=1^n} \leq C\lambda^{|i-j|}$$

where:

$$C = C(a,r) = \max\{a^{-1}, C_0\}.$$

Proof

Since A is positive definite and invertible we have $0 < a < b$ and we know that A is centered. Thus A^k is centered and km-banded for $k \geq 0$. Thus if p is a polynomial in $\mathbb{P}_{\leq k}$ then $p(A)$ is km-banded and centered. By the lemma (4) we know there exists a sequence of polynomials $\{p_n\}_{n \geq 1}$ in $\mathbb{P}_n$ satisfying:

$$\left\| \frac{1}{x} - p_n \right\|_{[a,b]} = C_0 q^{n+1}$$

Rewriting it for the matrix we have that:

$$\|A^{-1} - p_n(A)\| = \left\| \frac{1}{x} - p_n \right\|_{\sigma(A)} \leq C_0 q^{n+1}.$$

And now rewriting $|i-j| = nm/2 + k$ for $k = 1, 2, \ldots, m/2$ and $i \neq j$, we have that the inequality:

$$\frac{2|i-j|}{m} \leq (n+1)$$

holds, and hence we have that:

$$|a_{i,j}^{-1}| = |a_{i,j}^{-1} - p_n(a_{i,j})| \leq \|A^{-1} - p_n(A)\| \leq C_0 \lambda^{|i-j|}.$$

In case $i = j$ note that $a^{-1} = \|A^{-1}\|$, and this completes the proof.

Now following the authors we report the extension of the result for a more generic type of matrix A. Before doing this we need to define the *quasi-centered* matrix, we says that A is a *quasi-centered* if the central diagonal is contained within the nonzero bands of the matrix, i.e. $A \in \mathcal{B}(l^2(S))$ is invertible only if A is quasi-centered, note also that this is not true for $A \in l^2(Z)$.

Theorem 6

Let A be m-banded, bounded and boundedly invertible on $l^2(S)$. Let $[a, b]$ be the smallest interval containing $\sigma(AA^H)$. Then setting $r = b/a$, $q = q(r)$ as in the lemma (4), and $\lambda_1 = q^{1/m}$, there is a constant C_1 depending on A so that:

$$(|A^{-1}|)_{i,j=1}^{n} = |a_{i,j}^{-1}| \leq C_1 \lambda_1^{|i-j|}.$$

If A is quasi-centered then we may choose

$$C_1 = (m+1)\|A\|\lambda_1^{-m} C(a, r)$$

Proof

The results follows immediately from the previous proposition observing that:

- $A^{-1} = A^H (AA^H)^{-1}$;
- $\|A\| = \|A^H\|$.

Then the proof of the theorem (3) is given by two precedent proposition.

Sparse inverting the LU Factors

In this section we are going to present an idea by van Duin (1999), for obtaining an *explicit* preconditioner. The strategies proposed is performing a sparse inversion technique on the triangular factors of an *implicit* preconditioner. We are going to start from a sparse approximate LDU-factorization. After having obtained a sparse approximation for the matrices L^{-1} and U^{-1} we use their sparse inversion as the factor for an *explicit* preconditioner of the form $M^{-1} = \tilde{U}^{-1}D^{-1}\tilde{L}^{-1}$.

To reproduce this sparse inversion strategy we start expressing the U matrix as:[11]

$$U = I + \sum_{i=1}^{n-1} \mathbf{e}_i \mathbf{u}_{i,:}^T,$$

now observing that $\forall\, j \leq k$ we have $\mathbf{e}_k \mathbf{u}_k^T \mathbf{e}_j \mathbf{u}_j^T = 0$, since the j-th entry of $\mathbf{u}_k$ is zero $\forall\, j \leq k$, we can rewrite U as:

$$U = \prod_{i=n-1}^{1} \left(I + \mathbf{e}_i \mathbf{u}_i^T \right). \tag{3}$$

Now we can construct the inverse of the element in equation (3)[12]:

$$\left(I + \mathbf{e}_i \mathbf{u}_i^T \right)^{-1} = I - \mathbf{e}_i \mathbf{u}_i^T,$$

and then we have that:

$$U^{-1} = \prod_{i=1}^{n-1} \left(I - \mathbf{e}_i \mathbf{u}_i^T \right). \tag{4}$$

Now, since U^{-1} is also an upper triangular matrix, we could rewrite the expression as sum:

$$U^{-1} = I + \sum_{i=1}^{n-1} \mathbf{e}_i \hat{\mathbf{u}}_i^T. \tag{5}$$

where the $\hat{\mathbf{u}}_i^T$, the strictly upper triangular part of the i-th row of U^{-1}, is obtained as:

$$\hat{\mathbf{u}}_i^T = -\mathbf{u}_i^T \prod_{j=i+1}^{n-1} \left(I - \mathbf{e}_i \mathbf{u}_i^T \right). \tag{6}$$

The expression for the L^{-1} matrix can be obtained in a similar way[13] A straightforward implementation of the formula (6) is in the algorithm (1), that has the flaw of generating dense matrix, as we have already observed with the corollary (2). To account for this we have to generate some sparsification strategy via dropping, in the same manner we have done it for the incomplete LU factorization. Being analogue to the other technique we will give a general overview of them illustrating the modification of the algorithm (1).

Final value dropping Setting a drop tolerance ε for the value of α, the updating of the vector $\hat{\mathbf{u}}_i^T$ is done if the absolute value of α is greater than the tolerance ε.

[11] As usual we are using $\mathbf{e}_i$ notation for the vectors of the canonical basis, while $\mathbf{u}_i$ is the i-th row of the matrix U with the element $u_i(j) = 0$ for $j \leq i$.

[12] We are using the Sherman-Morrison formula for the inversion of the expression like $(A + \mathbf{u}\mathbf{v}^T)$, information are in (Sherman and Morrison, 1950).

[13] From the formula (6) we can observe that no $\hat{\mathbf{u}}_j$ is needed for the calculation of $\hat{\mathbf{u}}_i$ for $i \neq j$, so the whole inversion process can be executed in parallel on a distributed memory machine, you can see again the implementation details in (Bertaccini and Filippone, 2016).

Algorithm 1: Sparse product algorithm.

Input: $U \in \mathbb{R}^{n\times n}$ strict upper triangular matrix

for $i = 1, \dots, n-1$ **do**
  $\hat{\mathbf{u}}_i^T \leftarrow -\mathbf{u}_i^T$;
  $j \leftarrow$ first non-zero position in $\hat{\mathbf{u}}_i^T$;
  while $j < n$ **do**
    $\alpha \leftarrow -\hat{\mathbf{u}}_i^T \mathbf{e}_j$;
    $\hat{\mathbf{u}}_i^T = \hat{\mathbf{u}}_i^T + \alpha \mathbf{u}_j^T$; `// As a sparse operation.`
    $j \leftarrow$ next non-zero position in $\hat{\mathbf{u}}_i^T$;

Algorithm 2: Vector update drop strategy.

for $\{k \,|\, \boldsymbol{u}_j^T(k) \neq 0\}$ **do**
  $\mathbf{u} \Leftarrow \mathbf{u}_j^T(k)$;
  $d = \alpha \cdot \mathbf{u}$;
  if *Position k is not filled in* $\hat{\boldsymbol{u}}_i^T$ **then**
    if $|d| > \varepsilon$ **then**
      $\hat{\mathbf{u}}_i^T(k) = d$;
  else
    $\hat{\mathbf{u}}_i^T(k) = \hat{\mathbf{u}}_i^T(k) + d$;

Vector update dropping The fill-in is dropped as soon as it occurse in the sparse vectore update, see algorithm (2). [h]

Pattern drop A fixed pattern S for the matrix is given, so $\hat{\mathbf{u}}_i^T(k)$ is only calculated when $(i,k) \in S$.

Neumann drop [14] We start from a rewriting of othe formula (4), namely the Neumann series expansion for the formula (3):

[14] Note that the first two term are available without cost.

$$\begin{aligned} U^{-1} =& I - \sum_{j_1=1}^{n-1} \mathbf{e}_{j_i} \mathbf{u}_{j_1}^T + \sum_{j_2=1}^{n-2} \left(\mathbf{e}_{j_2} \mathbf{u}_{j_2}^T \sum_{j_1=j_2+1}^{n-1} \mathbf{e}_{j_1} \mathbf{u}_{j,1}^T \right) + \\ & - \sum_{j_3=1}^{n-3} \left(\mathbf{e}_{j_3} \mathbf{u}_{j_3}^T \sum_{j_2=j_3+1}^{n-2} \left(\mathbf{e}_{j_2} \mathbf{u}_{j_2}^T \sum_{j_1=j_2+1}^{n-1} \mathbf{e}_{j_1} \mathbf{u}_{j,1}^T \right) \right) + \dots \end{aligned} \tag{7}$$

by truncating this expression at a number of extra term m we obtain the dropping $\hat{U}_m$, the backside of this approach is the m-time computation of the update $\mathbf{u}_k^T$, in the worst case, to update $\hat{\mathbf{u}}_i^T$.

Positional fill level Similarly to the $ILU(P)$ we define a level of fill initialized for U as:

$$\mathrm{lev}_{i,j} = \begin{cases} 0 & \text{if } \mathbf{u}_i^T(j) \neq 0 \\ +\infty & \text{if } \mathbf{u}_i^T(j) = 0, \end{cases}$$

and the function to update the fill levels is:

$$\mathrm{lev}_{i,k} = \min(\mathrm{lev}_{i,j} + 1, \mathrm{lev}_{i,k}),$$

in this way the algorithm (1) becomes the algorithm (3).

Algorithm 3: Positional fill level inversion of a sparse triangular matrix

Input: $U \in \mathbb{R}^{n\times n}$ strict upper triangular matrix, inizial pattern of the matrix $\text{lev}_{i,j}$.

```
for j = 1, ..., n − 1 do
    û_i^T ← −u_i^T;
    j ← first non-zero position in û_i^T;
    while j < n do
        if lev_{i,j} ≤ p then
            α ← −û_i^T e_j;
            û_i^T ← û_i^T + α û_j^T;
            lev_{i,k} = min(lev_{i,j} +1, lev_{i,k});
        else
            û_i^T(j) ← 0;
        j ← next non-zero position in û_i^T;
```

Positional fill level II instead of using the level of fill of the approximate inverse matrix one can choose the level of fill of the original sparse triangular factor, this choice changes only the inizialization step, namely it becomes:

$$\text{lev}_{i,j} = \begin{cases} \text{lev}^U_{i,j} & \text{if } \mathbf{u}_i^T(j) \neq 0 \\ +\infty & \text{if } \mathbf{u}_i^T(j) = 0, \end{cases}$$

for the rest of the algorithm does not change from the algorithm (3).

Hybrid strategy In this way the algorithm (3) is combined with the drop strategies relative to the value of α, morally speaking the complete analogue of the ILUT(p,τ) algorithm.

References

Daniele Bertaccini and Salvatore Filippone. Sparse approximate inverse preconditioners on high performance gpu platforms. *Computers & Mathematics with Applications*, 71(3):693 – 711, 2016. ISSN 0898-1221. DOI http://dx.doi.org/10.1016/j.camwa.2015.12.008. URL http://www.sciencedirect.com/science/article/pii/S0898122115005763.

Michele Benzi and Miroslav Tuma. A comparative study of sparse approximate inverse preconditioners. *Applied Numerical Mathematics*, 30(2):305–340, 1999.

D. Bertaccini and F. Durastante. *Iterative methods and preconditioning for large and sparse linear systems with applications*. Chapman & Hall, 2017. In Preparation.

John R Gilbert. Predicting structure in sparse matrix computations. *SIAM Journal on Matrix Analysis and Applications*, 15(1):62–79, 1994.

Stephen Demko, William F Moss, and Philip W Smith. Decay rates for inverses of band matrices. *Mathematics of computation*, 43(168):491–499, 1984.

Victor Eijkhout and Ben Polman. Decay rates of inverses of banded m-matrices that are near to toeplitz matrices. *Linear Algebra and its Applications*, 109, 1988. DOI 10.1016/0024-3795(88)90211-x.

Reinhard Nabben. Decay rates of the inverse of nonsymmetric tridiagonal and band matrices. *SIAM Journal on Matrix Analysis and Applications*, 20, 1999. DOI 10.1137/s0895479897317259.

Claudio Canuto, Valeria Simoncini, and Marco Verani. On the decay of the inverse of matrices that are sum of kronecker products. *Linear Algebra and its Applications*, 452:21–39, 2014.

Pafnutï Lvovitch Tchebychev. Sur les polynômes réprésentant le mieux les valeurs des fonctions fractionnaires élémentaires pour les valeurs de la variable contenues entre deux limites données. In St. Petersburg, editor, *Oeuvres*, volume II, pages 669–678. 1907.

SN Bernstein. *Leçons sur les propriétés extrémales et la meilleure approximation des fonctions analytiques d'une variable réelle*. Paris, 1926.

Günter Meinardus and Larry L. Schumaker. *Approximation of functions: theory and numerical methods*. Springer Tracts in Natural Philosophy. Springer, 1 edition, 1967. ISBN 9783540039853,3540039856.

Arno C. N. van Duin. Scalable parallel preconditioning with the sparse approximate inverse of triangular matrices. *SIAM J. Matrix Anal. Appl.*, 20, 1999. DOI 10.1137/s0895479897317788.

Jack Sherman and Winifred J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Ann. Of Math. Stat.*, 21, 03 1950. DOI 10.2307/2236561.