



Wiedemann algorithm: an efficient approach for solving linear systems with sparse matrices

Moumouni DJASSIBO Woba^{1*}, ZONGO Moumouni², ZOUNGRANA Amidou²

¹Université Lédéa Bernard OUEDRAOGO (B.F).

²Université Norbert ZONGO, (B.F).

*Corresponding Author

Moumouni DJASSIBO Woba

Université Lédéa Bernard OUEDRAOGO (B.F).

Article History

Received: 25.01.2025

Accepted: 10.03.2025

Published: 30.03.2025

Abstract: Sparse matrices are matrices that contain many zero elements. The Wiedemann algorithm is an iterative method for solving linear systems represented by sparse matrices. It reduces the solution to the computation of the minimal polynomial, which itself relies on the recognition of a recurrent sequence with constant coefficients.

Keywords: Sparse matrices, Linear systems, Minimal polynomial, Recurrent sequence, Constant coefficients.

Cite this article:

Moumouni, D. W., Moumouni, Z., Amidou, Z., (2025). Wiedemann algorithm: an efficient approach for solving linear systems with sparse matrices. *ISAR Journal of Multidisciplinary Research and Studies*, 3(3), 36-38.

1. Introduction

In this article, we address radically different questions from dense linear algebra: we do not seek to perform products, inversions, etc., of arbitrary matrices, but to manipulate sparse matrices.

We will not begin by precisely defining what a sparse matrix is. The approach consists of providing algorithms whose complexity is expressed in terms of the number of non-zero elements in the input matrices. The complexity of the 'sparse' algorithms will yield a bound, typically on the order of

$O(n)$ for matrices of size

$n \times n$, regarding the number of non-zero coefficients for the model change to be relevant.

In what follows, we will therefore consider a matrix A of size

$n \times n$ with coefficients in a field

K satisfying the following assumptions:

- A is invertible,
- A contains s non-zero elements.

The product of A by a vector can thus be performed in

$O(s)$ operations in K . We will describe an algorithm due to Wiedemann that allows us to compute the unique solution of the system $Ax = y$ with a complexity of $O(ns)$.

The Wiedemann algorithm is more general than the one presented here, as it can handle rectangular or square non-invertible matrices, but it reduces to the case of invertible square matrices.[1]

Note that if s is on the order of n^2 , characterizing rather dense matrices, we revert to a complexity on the order of $O(n^3)$. The interesting case is when s is on the order of n , in which case the algorithm is quadratic in n : to simplify, we will note that the exponent of sparse linear algebra is 2 in favorable cases. [2]

2. Minimal polynomial and solution of systems

The Wiedemann algorithm involves calculating the minimal polynomial of A . Let us recall its definition. It is possible to associate with any polynomial in one variable

$$P = \sum_i p_i X^i \text{ de } K[X] \quad (1)$$

A matrix $P(A) = \sum_i p_i A^i$.

The Cayley–Hamilton theorem states that the characteristic polynomial χ_A of A annihilates the matrix A , meaning that $\chi_A(A) = 0$. [3]

Definition 2.1

The minimal polynomial of A is the monic polynomial of the smallest degree that annihilates A . Let $\mu_A(A)$ be the minimal

polynomial of A , assumed to be known. Since we have assumed A is invertible, the constant term of $\mu_A(A)$ is non-zero: it divides the constant term of the characteristic polynomial, which is also non-zero. The equation $\mu_A(A) = 0$ can be rewritten in the form:

$$A^{-1} = -\frac{1}{p_0}(A^{m-1} + p_{m-1}A^{m-2} + \dots + p_1I_n) \quad (2)$$

To solve the system

$Ax = b$ (with $b \in K^n$), we will calculate $x = A^{-1}b$, that is :

$$-\frac{1}{p_0}(A^{m-1}b + p_{m-1}A^{m-2}b + \dots + p_1b)$$

Thus, it is sufficient to compute the iterates $b, Ab, \dots, A^{m-1}b$, and then add the vectors $p_1b, p_2Ab, \dots, p_{m-1}b$ to obtain x . The complexity analysis is straightforward:

- Each $A^i b$ is obtained from $A^{i-1}b$ in $O(s)$ operations
- Each multiplication by p_i , and each addition, is done in $O(n)$ operations.

Note that $n \leq s$ (the invisibility assumption of A).

In total, we therefore have $O(ns)$ operations to perform to solve the system, provided we know the minimal polynomial of A . [4]

3. Calculation of the minimal polynomial

Let us write the minimal polynomial of A in the form

$$\mu_A(X) = X^m + p_{m-1}X^{m-1} + \dots + p_0. \quad (3)$$

Then the sequence of powers of A satisfies the linear recurrence:

$$A^{k+m} + p_{m-1}A^{k+m-1} + \dots + p_0A^k = 0$$

Input: A sparse matrix $A \in M_n(K)$.

Output: Its minimal polynomial $\mu_A(X)$.

1. Choose random vectors u and v in K^n .
2. For $0 \leq i \leq 2n$, calculate the vectors $v_i = A^i v$, then the scalars $N_i = u^T v_i$.
3. Return the minimal polynomial of the sequence N_i . [5]

This sequence does not satisfy any recurrence of smaller order. For all vectors u and v , the sequence of numbers $N_i := u^T v_i$ therefore satisfies:

$$N_{k+m} + p_{m-1}N_{k+m-1} + \dots + p_0N_k = 0 \quad (4)$$

If u and v are poorly chosen (for example, null vectors), the sequence

N_i satisfies a recurrence of order smaller than m . The following proposition shows that by randomly choosing u and v we are likely to obtain a sequence

N_i that does not satisfy a recurrence of smaller order.

Proposition 3.1

There exists a non-zero polynomial

$D \in K[U_1, \dots, U_n, V_1, \dots, V_n]$ of degree at most $2n$, such that if $D(u, v)$ is non zero, the sequence N_i associated with u and v does not satisfy a recurrence of order smaller than m .

The idea is to choose u and v randomly. If we are unlucky, $D(u, v)$ is zero; otherwise, the sequence of N_i associated with u and v allows us to recover $\mu_A(X)$ using the Berlekamp–Massey algorithm. The algorithm summarizes this probabilistic approach. The complexity of step (2) is $O(n)$ matrix-vector or vector-vector products, each taking at most $O(s)$ operations; in total, we thus obtain

$O(ns)$ operations for this phase. The Padé approximant algorithm is negligible, as it requires only $O(n^2) = O(sn)$ opérations.[6]

4. Calculation of the Determinant

The algorithm allows for detecting the invertibility of the matrix A during the computation. Indeed, if the Padé approximant computed by the Berlekamp–Massey algorithm has a denominator divisible by X , it means that $\det(A) = 0$.

It is possible to compute the determinant of the sparse matrix A efficiently, by exploiting the following result from linear algebra.

Lemma 4.1: If all the principal minors of a matrix A are non-zero, then the minimal and characteristic polynomials of a matrix $B = A \cdot \text{Diag}(X_1, \dots, X_n)$ coincide.

The algorithm that follows from this is as follows:

- Choose a random permutation matrix P to ensure that all the principal minors of P are invertible.
- Select elements $x_1, \dots, x_n$, randomly from a field K .
- Compute the minimal polynomial μ_B .
- Return $\frac{\mu_B(0)}{(x_1, \dots, x_n)} \text{sgn}(P)$.

The algorithm has a quadratic complexity with respect to the size n of A . [7]

5. Calculation of the Rank

The maximum rank of the sparse matrix A is detected by the previous algorithms. It is shown that for a random choice of elements in K , the rank r is equal to $\deg(u_{AD}) - 1$, where D is the diagonal matrix $D = \text{Diag}(x_1, \dots, x_n)$.

- The indication for the proof requires the use of a resultant.
- One point to demonstrate is that the polynomial R has at most $d \cdot \text{card}(U)^{n-1}$ roots in U^n .
- This leads to the conclusion that for an element of U^n whose coordinates are chosen randomly from UUU , the probability of being a root of R is bounded by $\frac{d}{\text{card}(U)}$.
- In conclusion, it is established that the probability P satisfies

$$P \leq 1 - \frac{d}{\text{card}(U)}.$$

6. Conclusion

The Wiedemann algorithm represents a significant advancement in the processing of linear systems represented by sparse matrices. By exploiting the particular structure of these matrices, the algorithm performs calculations with reduced complexity, thereby enabling the solution of problems that would be intractable with classical methods for dense matrices.

This article has presented the fundamental principles of the algorithm, including the computation of the minimal polynomial and its use in solving linear systems. The theoretical results, such as the relationship between the roots of the polynomial and the rank of the matrix, provide insights into the probabilities of invertibility and the detection of singularities.

The practical implications of these methods are vast, affecting various fields such as cryptography, optimization, and data processing. As the applications of sparse matrices multiply, the Wiedemann algorithm and its variants will continue to play an essential role, facilitating the development of efficient and robust algorithms for numerical computation.

It is therefore crucial to continue research in this area, particularly to further improve the efficiency of the algorithms and explore new potential applications.

Bibliography

- [1] Coppersmith, Don (1994). «Solving homogeneous linear equations over GF (2) via block Wiedemann algorithm». In : *Mathematics of Computation*, vol. 62, n205, p. 333–350.
- [2] DeMillo, Richard A. et Richard J. Lipton (1978). «A probabilistic remark on algebraic program testing». In: *Information Processing Letters*, vol. 7, n4, p. 193–195.
- [3] Giesbrecht, Mark (2001). «Fast computation of the Smith form of a sparse integer matrix ». In: *Computational Complexity*, vol. 10, n1, p. 41–69.
- [4] Giesbrecht, M., A. Lobo et B. D. Saunders (1998). «Certifying inconsistency of sparse linear systems ». In : *ISSAC'98 : International Symposium on Symbolic and Algebraic Computation*. New York: ACM Press, p. 113–119.
- [5] Kaltofen, Erich et B. David Saunders (1991). «On Wiedemann's method of solving sparse linear systems ». In : *Applied algebra, algebraic algorithms and error-correcting codes* (New Orleans, LA, 1991). Vol. 539. *Lecture Notes in Computer Science*. Springer-Verlag, p. 29–38.
- [6] Mulders, Thom (2004). «Certified sparse linear system solving ». In: *Journal of Symbolic Computation*, vol. 38, n5, p. 1343–1373.
- [7] Schwartz, J. T. (1980). « Fast probabilistic algorithms for verification of polynomial identities ». In: *Journal of the Association for Computing Machinery*, vol. 27, n4, p. 701–717