Block solution of sparse linear systems over GF(q): the singular case

G. Villard

LMC-IMAG, B.P. 53 F38041 GRENOBLE cedex 9

Gilles.Villard@imag.fr

Abstract

Probabilistic algorithms based on the Krylov / Wiedemann or the Lanczos method to solve non homogeneous $N \times N$ systems $Ax = b$ over a Galois field GF(q), usually require $2N$ matrix times vector products and $O(n^{2+o(1)})$ additional arithmetic operations. Only the block Wiedemann algorithm as given by Kaltofen, has the least number $(1+\epsilon)N + O(1)$ of matrix-vector products of any known algorithm. We extend its analysis to the case of singular matrices A.

Introduction

We consider the problem of solving large unstructured sparse linear $N \times N$ systems of equations $Ax = b$ over Galois fields K =GF(q). Such systems can be solved by using structured Gaussian elimination. Another approach is to apply either the conjugate gradient, the Lanczos or the Krylov / Wiedemann method. It is well known that these latter methods are strongly related to each other. For a unified theory in the finite field case, the reader may refer to [7].

Typical in numerical analysis, these techniques are not yet fully analyzed over algebraic domains. Proposed and experimented as heuristics in [6, 1, 2, 8], they lead to empirical data, which point out the lack of theoretical explanations for various problems. Attempts of probability analysis are made in [8, 7] but only for random inputs.

The Wiedemann approach is the best understood technique. To *find dependencies i.e.* to solve $Ax = 0$ or to *solve nonsingular systems* $Ax = b$, the corresponding algorithms are rigorously proven to work with probability strictly greater than zero over any field. We refer to [12] for the non block method and to [4, 9, 11] for the block method of Coppersmith [2]. Note that, using the unified approach of [7], these results may be extended to the conjugate gradient and to the Lanczos non block methods.

These studies lead to the following cost analysis. Over sufficiently large fields, the least number of matrix-vector products, required to solve a system $Ax = b$ for any known algorithm, is $(1+\epsilon)N + O(1)$, for any arbitrarily small positive constant ϵ. It can be arrived at, only using the block method in [4]. Unfortunately its full analysis was restricted to invertible matrices.

Indeed, the case $Ax = b$ with A singular is a little bit more difficult. The solution proposed by Wiedemann involves $O(N \log N)$ matrix-vector products over any field [12]. This can be improved to $O(N)$ products using the randomization in [5] and to $2N$ products using the results in [3]. This latter paper is about the Lanczos algorithm, but its conclusions carry over to the Wiedemann one. Any system $Ax = b$ over a large cardinality field, can thus be solved with high probability using $2N$ matrix-vector products and $O(N^2)$ additional arithmetic operations.

In this short note we show that some ideas used to obtain the two latter results, for general non homogeneous systems, can be applied to the block Wiedemann algorithm of Coppersmith [2]. To use the block approach will allow us to reduce the number of required matrix-vector products. Precisely, this will lead to the least known number (given above) of products for such methods. Any system $Ax = b$ over a sufficiently large field can be solved, with high probability, in $(1+\epsilon)N + O(1)$ matrix-vector products and $O(N^2 \log N \log\log N)$ additional arithmetic operations.

This new bound is obtained using arguments which have implications for the understanding of the block methods in general.

The result is established in §2, we begin in §1 with a short presentation of the algorithm. To limit the length of the presentation, we assume the reader is a little bit familiar with the papers [12, 4] and [11].

1 Block solution of non homogeneous systems

We adapt the Coppersmith's version [2] of Wiedemann's algorithm to the solution of a non homogeneous system $Ax = b$. We use the terminology proposed in [11].
The algorithm picks up two random matrices X in $\mathcal{M}_{m,N}(K)$ and Y in $\mathcal{M}_{N,n-1}(K)$, then computes $Z = [b, AY]$.

For two integers δ_l and δ_r to be determined in next section, one next generate the matrix sequence

$$H_i = XA^iZ, \quad i = 0, \ldots, \delta_l + \delta_r - 1.$$

The solution is computed from a matrix generating polynomial [11] for the sequence $\{H_i\}_i$. We mean a nonsingular polynomial matrix

$$D_Z(\lambda) = D_0 + D_1\lambda + D_2\lambda^2 + \ldots + D_d\lambda^d \quad \in \ \mathcal{M}_{n,n}(K[\lambda])$$

with $d \le \delta_r$, such that

$$XA^iZD_0 + XA^{i+1}ZD_1 + \ldots + XA^{i+d}ZD_d = 0$$

for $0 \leq i \leq \delta_l - 1$.

As we will prove later, with high probability, the left projection by X does not modify the invariants of the sequence and $D_Z(\lambda)$ is also a matrix generating polynomial for $\{A^i Z\}_i$:

$$A^i Z D_0 + A^{i+1} Z D_1 + \ldots + A^{i+d} Z D_d = 0 \qquad (1)$$

for $0 \leq i \leq \delta_l - 1$.

Further, up to a preconditioning of A (see next section), with high probability, $D_Z(\lambda)$ will also satisfy:

$$(\det D_Z(\lambda))(0) \neq 0. \qquad (2)$$

Thus one can find a permutation matrix P in $\mathcal{M}_{n,n}(K)$ such that

$$(D_Z(\lambda)Q)(0) = T$$

where the first entry $T_{1,1}$ of T is nonzero in K. Let the first column of $G_Z(\lambda) = D_Z(\lambda)Q$ be denoted by

$$G^{(1)} = {}^t[g_1(\lambda), g_2(\lambda), \ldots g_n(\lambda)]$$

we know that

$$g_1(0) \neq 0$$

and obviously $G_Z(\lambda)$ is a generating polynomial having the same properties than $D_Z(\lambda)$. Now, using the first columns of the matrices in (1) with $i = 0$ and denoting the last $n-1$ columns of Z by $Z_2, \ldots, Z_n$:

$$g_1(A)b + g_2(A)Z_2 + \ldots + g_n(A)Z_n = 0.$$

If $g_1(\lambda) = g_1(0) + \lambda h(\lambda)$ and the columns of Y are $Y_2, \ldots, Y_n$, we finally get:

$$-g_1(0)b = A\left(h(A)b + g_2(A)Y_2 + \ldots + g_n(A)Y_n\right).$$

The algorithm may thus output as a solution x for the system $Ax = b$:

$$x = -1/g_1(0)\left(h(A)b + g_2(A)Y_2 + \ldots + g_n(A)Y_n\right).$$

The algorithm is randomized concerning two points: to satisfy (2) we rely below on results in [12, 5]. To ensure that the left projection is correct we use our previous studies of the algorithm in the homogeneous case [9, 11].

2 Probabilistic analysis

We first give the necessary conditions to satisfy (2). Then we precise which values of δ_l and δ_r should be chosen to characterize to required sequence length in the algorithm. We conclude with the main result giving the overall cost.

Next two lemmata are consequences of corollary 1 in [11] or of corollary 6.5 in [9]. The first lemma can also be directly shown by classical means of linear algebra, we omit its proof.

Lemma 1 *Suppose A in $\mathcal{M}_N(K)$ with rank r is similar to a block-diagonal matrix* $\operatorname{diag}(B, 0)$ *where B is invertible and has rank r as well. Then if $Z = AY$, the minimal matrix generating polynomial $D_Z(\lambda)$ for the sequence $\{A^i Z\}_{i=0}^{\infty}$ is such that* $(\det D_Z)(0) \neq 0$.

In addition, using blocking matrices with indeterminate entries, we can bound the degree of the matrix minimal polynomial in the generic cases. Let $\mathcal{X}$ and $\mathcal{Y}$ be matrices which entries are indeterminates $\xi_{j,k}$, $1 \leq j \leq m$, and $\upsilon_{k,l}$, $1 \leq l \leq n-1$, with $1 \leq k \leq N$.

Let A^* be a restriction of A to its range space. Suppose that the Frobenius form of A^* has ϕ^* companion blocks. Let ν^* be the sum of the dimensions of the first $\min\{\phi^*, n\}$ blocks. From [11], with a right blocking matrix with n columns, we know that the degree of the minimal polynomial depends on ν^*, we deduce that:

Lemma 2 *Let $\mathcal{Z} = A\mathcal{Y}$ and $\mathcal{Z}_b = [b,\ \mathcal{Z}]$ and suppose A satisfies the conditions of lemma 1. If $m \geq \min\{\phi^*, n\}$, the minimal polynomial $D_{\mathcal{Z}_b}(\lambda)$ for $\{A^i \mathcal{Z}_b\}_{i=0}^{\infty}$ is equal to the one for $\{\mathcal{X} A^i \mathcal{Z}_b\}_{i=0}^{\infty}$. Its degree is less than $\delta_r = \lceil \nu^*/(n-1) \rceil$. It can be computed from the sequence $\{\mathcal{X} A^i \mathcal{Z}_b\}_{i=0}^{\delta_l+\delta_r-1}$ where $\delta_l = \lceil N/m \rceil$.*

Proof. Applying proposition 2 of [11] on the left we know that the two minimal polynomials are equal. We now bound the degree of the minimal polynomial for $\{\mathcal{X} A^i \mathcal{Z}_b\}_{i=0}^{\infty}$ from a bound for the one for $\{\mathcal{X} A^i \mathcal{Z}\}_{i=0}^{\infty}$. From corollary 1 in [11], the minimal polynomial for $\{A^i \mathcal{Z}\}_{i=0}^{\infty}$ is equal to the one for $\{\mathcal{X} A^i \mathcal{Z}\}_{i=0}^{\infty}$. It has degree exactly $\lceil \bar{\nu}^*/(n-1) \rceil$ if $\bar{\nu}^*$ is the sum of the dimensions of the first $\min\{\phi^*, n-1\}$ blocks of the Frobenius form of A^*. By definition of $\bar{\nu}^*$, $\lceil \bar{\nu}^*/(n-1) \rceil \leq \lceil \nu^*/(n-1) \rceil$. Since, by definition of ν^*, the n-th block of the latter Frobenius form has dimension less than $\lceil \nu^*/n \rceil$ which is also less than $\lceil \nu^*/(n-1) \rceil$, the bound holds for the degree of the minimal polynomial for $\{\mathcal{X} A^i \mathcal{Z}_b\}_{i=0}^{\infty}$. The bound on δ_l directly comes from corollary 6.5 in [9]. □

We can now prove the final result. This is an extension of the corollary of page 795 in [4] for general matrices.

Theorem 1 *Let A be an $N \times N$ matrix over $K = GF(q)$. Let $\epsilon > 0$ be fixed. Suppose that X with m rows and Y with $n-1$ columns are chosen at random over K. Let b be in the range space of A. Then, using a preconditioning of A, the block algorithm of section 1 computes a solution x such that $Ax = b$, in no more than $(1+\epsilon)N + O(1)$ matrix-vector products of dimension N and $O(N^2 \log N \log\log N)$ additional operations in K. A solution is produced with probability greater than $1 - O(N^2)/(\text{card } K - 1)$.*

Proof. Let us first show that the algorithm produces a solution with the claimed probability. To ensure that $(\det D_Z(\lambda))(0) \neq 0$ we apply lemma 1. The condition on A is satisfied by preconditioning A using Toeplitz matrices (theorem 2 in [5]) and a diagonal matrix (lemma 2 in [5]) as done in [4]. The condition on Z is satisfied by construction and by assumption on b. Next we ensure that the left projection using X does not modify the matrix generating polynomials. We use the result given by lemma 2 in the generic case and get the target probability for random X and Y using either theorem 6 in [4] or theorem 5 in [11]. The condition on m is satisfied since the above preconditioning ensures that $\phi^* = 1$ [5].

We now detail the cost of the algorithm. Including the cost of the preconditioning, multiplying A by a vector requires one product of a matrix times a vector of dimensions N plus $O(\log N \log\log N)$ arithmetic operations in K [4]. With $\delta_l \geq N/m$ and $\delta_r \geq N/(n-1)$, $(n/(n-1) + n/m)N$ multiplications of A times a vector are required to generate the sequence $\{H_i\}_{i=0}^{\delta_l+\delta_r-1}$; $N/(n-1)$ additional multiplications plus $O(N^2)$ operations in K are needed to compute the solution x from a generating polynomial of degree less than $N/(n-1)$ [4]. The overall number of matrix-vector products is thus $(1 + n/m + 2/(n-1))N$, with $O((1 + n/m)N^2 \log N \log\log N)$ additional operations.

A matrix generating polynomial $D_Z(\lambda)$ is computed in $O((m+n)N^2$ arithmetic operations [2, 4, 10]. From there $G_Z(\lambda)$ is obtained at a lower cost.

Following [4] and formulating these costs in terms of $\epsilon = n/m + 2/(n-1)$, a quantity that can be made arbitrarily close to zero for well chosen values of m and n, the theorem is proven. □

Remark 1 *The preconditioning used for A, actually ensures stronger properties for the spectral structure of A than the one required in lemma 1. That why, as explained in the proof, there is no condition on m in the theorem. The preconditioning could thus be relaxed.*

Conclusion

We have proven that another application of scalar methods for the solution of sparse linear system, carries over to the block method of Coppersmith / Wiedemann. Future works will try to understand how the corresponding results can be applied to other block methods, especially to the Lanczos and the conjugate gradient ones.

References

[1] COPPERSMITH, D. Solving linear equations over GF(2): block Lanczos algorithm. *Linear Algebra and its Applications 192* (1993), 33–60.

[2] COPPERSMITH, D. Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm. *Math. Comp. 62*, 205 (1994), 333–350.

[3] EBERLY, W., AND KALTOFEN, E. On randomized Lanczos algorithms. In *International Symposium on Symbolic and Algebraic Computation, Maui, Hawaii, USA* (July 1997), ACM Press.

[4] KALTOFEN, E. Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. *Math. Comp. 64*, 210 (1995), 777–806.

[5] KALTOFEN, E., AND SAUNDERS, B. On Wiedemann's method of solving sparse linear systems. In *Proc. AAECC-9* (1991), LNCS 539, Springer Verlag, pp. 29–38.

[6] LAMACCHIA, B., AND ODLYZKO, A. Solving large sparse linear systems over finite fields. In *Advances in Cryptology – CRYPTO'90, Springer LNCS 537* (1991), A. Menezes and S. Vanstone, Eds., pp. 109–133.

[7] LAMBERT, R. *Computational aspects of discrete logarithms*. PhD thesis, University of Waterloo, Ontario, Canada, 1996.

[8] MONTGOMERY, P. A block Lanczos algorithm for finding dependencies over GF(2). In *EUROCRYPT'95, Heidelberg, Germany. Springer LNCS 921* (1995), pp. 106–120.

[9] VILLARD, G. A study of Coppersmith's block Wiedemann algorithm using matrix polynomials, Feb. 1997. RR 975-I-M IMAG Grenoble, France.

[10] VILLARD, G. Computing minimum generating matrix polynomials, 1997. Preprint IMAG Grenoble, France.

[11] VILLARD, G. Further analysis of Coppersmith's block Wiedemann algorithm for the solution of sparse linear systems. In *International Symposium on Symbolic and Algebraic Computation, Maui, Hawaii, USA* (July 1997), ACM Press.

[12] WIEDEMANN, D. Solving sparse linear equations over finite fields. *IEEE Transf. Inform. Theory 32* (1986), 54–62.