Ann. Acad. Rom. Sci.
Ser. Math. Appl.
ISSN 2066-6594 Vol. 8, No. 2/2016

On a direct solver for linear least squares problems*

Constantin Popa†

Abstract

The Null Space (NS) algorithm is a direct solver for linear systems of equations. It was initially designed and theoretically analyzed by M. Benzi in 1993 for square nonsingular systems, and its main idea consists on projections of an initial set of vectors onto the hyperplanes associated to the system equations, by using projections parallel with some specific directions which are constructed during the development of the algorithm. In this paper we extend and theoretically analyze the NS algorithm to linear least squares problems.

MSC: 65F05, 65F20

keywords: consistent linear systems; Null Space algorithm; linear least squares problems; normal equations; augmented system

1 Introduction

For $A : m \times n$ and $b \in I\!R^m$ we will consider the system of linear equations

$$Ax = b. \tag{1}$$

If b is in the range of A, i.e. at least one $z \in I\!R^n$ exists such that $Az = b$, we say that (1) is consistent and denote by $S(A; b)$ the set of all its solutions

*Accepted for publication in revised form on November 1-st 2016
†cpopa1956@gmail.com; cpopa@univ-ovidius.ro; Ovidius University of Constanta, Romania

and by x_{LS} the minimal norm one. If $Az \neq b$, $\forall z \in I\!R^n$ we say that (1) is inconsistent and reformulate it as a linear least-squares problem: find $x^* \in I\!R^n$ such that

$$\| Ax^* - b \| = \inf\{\| Ax - b \|, x \in I\!R^n\}. \tag{2}$$

Let $LSS(A; b)$ be the set of all its solutions and x_{LS} the (unique) solution of minimal norm. A first equivalent consistent formulation of the problem (2) is given by the associated normal equation (see e.g. [8]).

$$A^T A x^* = A^T b. \tag{3}$$

Unfortunately, in order to use this equation, the computation of the product $A^T A$ is required, and this is a very expensive procedure for both computational time and memory aspects. Fortunately, it exists a more convenient consistent equivalent formulation of (3) through the augmented system, i.e.

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}. \tag{4}$$

The equivalence of the problems (2) and (4) is shown in Proposition 1.2 from [7]. In [2] (see also [3]) the authors introduce and theoretically analyze the Null Space (NS) algorithm, as a direct method for numerical solution of square nonsingular systems of linear equations. According to these aspects, the paper is organized as follows: in section 2 we briefly describe the original NS algorithm from [2] - [3]. In section 3 we extend and theoretically analyze the algorithm NS to general rectangular, but consistent systems of linear equations and we point out how this extension can solve also inconsistent linear systems (linear least squares problems), through the equivalent formulations (3) - (4).

2 The Null Space algorithm

In this section we will briefly describe the original Null Space (NS) algorithm, introduced in [2] (see also [3]) for square **nonsingular** systems of linear equations of the form

$$Ax^* = b, \tag{5}$$

with $x^* = A^{-1}b$ its unique solution, where $A : n \times n$ and $b \in I\!R^n$. We shall denote by $\langle \cdot, \cdot \rangle, \| \cdot \|$ the Euclidean scalar product and norm, and by

$$S_i = \{x \in I\!R^n, \langle A_i, x \rangle = 0\}, \quad H_i = \{x \in I\!R^n, \langle A_i, x \rangle = b_i\} = S_i + \frac{b_i}{\| A_i \|^2} A_i, \tag{6}$$

the vector subspace and hyperplane, respectively associated to the i-th equation of the system (5), where $A_i, i = 1, \ldots, n$ are the rows of A. For $d \in I\!R^n$ such that $\langle d, A_i \rangle \neq 0$ and any $x \in I\!R^n$ there exists the projections of x parallel with d onto S_i and H_i (also called "directional projections"), defined by

$$P^d_{S_i}(x) = x - \frac{\langle x, A_i \rangle}{\langle d, A_i \rangle} d, \quad P^d_{H_i}(x) = x - \frac{\langle x, A_i \rangle - b_i}{\langle d, A_i \rangle} d. \tag{7}$$

For any $1 \leq k \leq n$, let $A^{(k)} : k \times n$ be the submatrix of A formed with its first k rows, i.e.

$$A^{(k)} = \begin{bmatrix} A_1^T \\ A_2^T \\ \cdots \\ A_k^T \end{bmatrix}, \tag{8}$$

and $\mathcal{N}_k$ the null space of $A^{(k)}$. The following relations are obvious

$$\mathcal{N}_k = S_1 \cap \cdots \cap S_k, \quad \{0\} = \mathcal{N}_n \subset \mathcal{N}_{n-1} \subset \cdots \subset \mathcal{N}_2 \subset \mathcal{N}_1 \tag{9}$$

and

$$dim(\mathcal{N}_k) = n - k, k = 1, \ldots, n. \tag{10}$$

In [2] (see also [3]) the authors consider a set of *null vectors*

$$\mathcal{N} = \{z^2, z^3, \ldots, z^n\} \subset I\!R^n \tag{11}$$

such that

$$z^k \neq 0, z^k \in \mathcal{N}_{k-1}, z^k \notin \mathcal{N}_k, \forall k = 2, \ldots, n, \tag{12}$$

i.e.

$$\langle z^k, A_j \rangle = 0, \forall j = 1, \ldots, k-1 \text{ and } \langle z^k, A_k \rangle \neq 0. \tag{13}$$

If such a set of null vectors (11)-(13) is available, we can compute x^* from (5) by the following direct solver (DS).

Algorithm DS.

Step 0 (Initialization): any vector $x \in H_1$

Step 1 (Successive projections):
for $i = 2 : n$

$$x = P^{z^i}_{H_i}(x) = x - \frac{\langle x, A_i \rangle - b_i}{\langle z^i, A_i \rangle} z^i$$

end

Remark 1. *We shall denote by* $e^1, e^2, \ldots, e^n$ *the canonical basis in* $I\!R^n$*. Then, a possible choice for* x *in the above Step 0 can be (see [3])*

$$x = \frac{b_1}{A_{1k_0}} e^{k_0}, \text{ where } A_{1k_0} \neq 0. \tag{14}$$

We reobtain the following result from [2].

Proposition 1. *The algorithm DS gives us the unique solution* x^* *from (5).*

Proof. We first observe that from (12)-(13) it results that

$$z^k \in S_1 \cap S_2 \cap \cdots \cap S_{k-1}, \tag{15}$$

$\forall k = 2, \ldots, n$. The statement of the proposition is equivalent with the following property

$$x^k \in H_1 \cap H_2 \cap \cdots \cap H_k, \tag{16}$$

$\forall 1 \leq k \leq n$, which we will prove by a recursive argument following k. For $k = 1$ we get that $x^1 \in H_1$ according to the construction in Step 0 of the algorithm DS. Then, let $n - 1 \geq k \geq 2$ be such that (16) holds for it. By using Step 1 of DS we get

$$x^{k+1} = P_{H_{k+1}}^{z^{k+1}}(x^k) = x^k - \frac{\langle x^k, A_{k+1} \rangle - b_{k+1}}{\langle z^{k+1}, A_{k+1} \rangle} z^{k+1} \in H_{k+1}. \tag{17}$$

From (16) (i.e $x^k \in H_j, j = 1, \ldots, k$), (15) (i.e. $z^{k+1} \in S_j, j = 1, \ldots, k$), the last equality in (6) and (17) (i.e. $x^{k+1} = x^k + \alpha_k z^{k+1}$ ($\alpha_k \in I\!R$ is the scalar from (17)) we get $x^{k+1} \in H_j, j = 1, \ldots, k$, which together with (17) completes the recursion argument and the proof. □

In order to construct the above set of null vectors, the authors proposed in [3] the following Null Vectors (NV) algorithm.

Algorithm NV.

Step 1 (Initialization): any basis $\mathcal{Z}_1 = \{z_2^{(1)}, \ldots, z_n^{(1)}\}$ in $\mathcal{N}_1$

Step 2 Because $dim(\mathcal{N}_2) = n - 2$ we must have $\langle A_2, z_i^{(1)} \rangle \neq 0$, for some $i \in \{2, 3, \ldots, n-1, n\}$, so we can permute the elements of $\mathcal{Z}_1$ such that $\langle A_2, z_2^{(1)} \rangle \neq 0$. Moreover, for the numerical stability of the algorithm we may choose

$$0 < |\langle A_2, z_2^{(1)} \rangle| = \max_{2 \leq i \leq n} |\langle A_2, z_i^{(1)} \rangle|. \tag{18}$$

Then, we produce the new set $\mathcal{Z}_2 = \{z_3^{(2)}, \ldots, z_n^{(2)}\}$ by using directional projections

$$z_j^{(2)} = P_{H_2}^{z_2^{(1)}}(z_j^{(1)}) = z_j^{(1)} - \frac{\langle z_j^{(1)}, A_2\rangle}{\langle z_2^{(1)}, A_2\rangle} z_2^{(1)}, \; j = 3, \ldots, n. \tag{19}$$

$$\vdots$$

Step n Because $dim(\mathcal{N}_{n-1}) = 1$ we must have $\langle A_{n-1}, z_i^{(n-2)}\rangle \neq 0$, for some $i \in \{n-1, n\}$, so we can permute the elements of $\mathcal{Z}_{n-2} = \{z_{n-1}^{(n-2)}, z_n^{(n-2)}\}$ such that $\langle A_{n-1}, z_{n-1}^{(n-2)}\rangle \neq 0$. Moreover, for the numerical stability of the algorithm we may choose

$$0 < |\langle A_{n-1}, z_{n-1}^{(n-2)}\rangle| = \max_{n-1 \leq i \leq n} |\langle A_{n-1}, z_i^{(n-2)}\rangle|. \tag{20}$$

Then, we produce the last set $\mathcal{Z}_{n-1} = \{z_n^{(n-1)}\}$ by

$$z_n^{(n-1)} = P_{H_{n-1}}^{z_{n-1}^{(n-2)}}(z_n^{(n-2)}) = z_n^{(n-2)} - \frac{\langle {}_n^{(n-2)}, A_{n-1}\rangle}{\langle z_{n-1}^{(n-2)}, A_{n-1}\rangle} z_{n-1}^{(n-2)}. \tag{21}$$

Step n+1 The set $\mathcal{Z}$ of null vectors is given by a last directional projection

$$\mathcal{Z} = \{z_2^{(1)}, z_3^{(2)}, \ldots, z_n^{(n-1)}\}. \tag{22}$$

Remark 2. *Because we start with a basis $\mathcal{Z}_1$ in $\mathcal{N}_1$, it can be proved that the vectors from $\mathcal{Z}_2, \ldots, \mathcal{Z}_{n-1}$ are also basis in $\mathcal{N}_2, \ldots, \mathcal{N}_{n-1}$, respectively (for the proof see [3]).*

Remark 3. *Following the ideas from [3], a basis $\mathcal{Z}_1 = \{z_2^{(1)}, \ldots, z_n^{(1)}\}$ in $\mathcal{N}_1$ can be constructed as indicated in the following Matlab code.*

```
Given A: n x n;
I=eye(n);z=zeros(n);
[A1k,k]=max(abs(A(1,:)));z(:,1)=I(:,k);
if (k==1)
   for i=1:n-1
    z(:,i+1)=I(:,i+1)-(A(1,i+1)/A(1,k))*I(:,k);
   end
else
```

```
    for i=1:k-1
    z(:,i+1)=I(:,i)-(A(1,i)/A(1,k))*I(:,k);
    end
   for i=k:n-1
    z(:,i+1)=I(:,i+1)-(A(1,i+1)/A(1,k))*I(:,k);
   end
end
```

We shall call the above NV algorithm together with the solution part DS the Null Space algorithm (NS). It can be written as follows.
Algorithm NS.

Step 1 (Initialization): any basis $\mathcal{Z}_1 = \{z^2, \ldots, z^n\}$ in $\mathcal{N}_1$

Step 2 for i=2:n-1

$maxim = \max_{i \leq j \leq n} |\langle A_i, z^j \rangle| = |\langle A_i, z^{j^*} \rangle|$.
$p(i) = \langle A_i, z^{j^*} \rangle$
if $(j^* > i)$ interchange the vectors z^i and z^{j^*}
for j=i+1:n
$z^j = P^{z^i}_{H_i}(z^j) = z^j - \frac{\langle z^j, A_i \rangle}{p(i)} z^i$
end

end

Step 3 (Initialization): any vector $x \in H_1$

Step 4 (Successive projections):
for $i = 2 : n$

$$x = P^{z^i}_{H_i}(x) = x - \frac{\langle x, A_i \rangle - b_i}{p(i)} z^i$$

end

3 The generalized Null Space algorithm

For A a general $m \times n$ matrix and $b \in \mathcal{R}(A) \subset I\!R^m$ we will consider the consistent linear system of equations

$$Ax = b. \tag{23}$$

Without restricting the generality of (23) we will suppose that the rows A_i and columns A^j of A satisfy the assumptions

$$A_i \neq 0,\ i = 1, \dots, m,\ A^j \neq 0,\ j = 1, \dots, n. \quad (24)$$

We propose for the numerical solution of the system (23) the following extension of the algorithm Null Space from the previous section.
Algorithm General Null Space (GNS)

Step 1 (Initialization): any basis $\mathcal{Z}_1 = \{z^2, \dots, z^n\}$ in $\mathcal{N}_1$; $\tau = 2$

Step 2 for i=2:m

- if $(maxim \neq 0)$
  - $maxim = \max_{\tau \leq j \leq n} |\langle A_i, z^j \rangle| = |\langle A_i, z^{j^*} \rangle|$.
  - $p(i) = \langle A_i, z^{j^*} \rangle$
  - if $(j^* > \tau)$ interchange the vectors z^τ and z^{j^*}
  - for j=τ+1:n
  - $z^j = P^{z^\tau}_{H_i}(z^j) = z^j - \frac{\langle z^j, A_i \rangle}{p(i)} z^\tau$
  - end
  - $\tau = \tau + 1$
- else $p(i) = 0$

end

Step 3 (Initialization): any vector $x \in H_1$; set $\tau = 2$

Step 4 for i=2:m

- if $(p(i) \neq 0)$
- $x = P^{z^\tau}_{H_i}(x) = x - \frac{\langle x, A_i \rangle - b_i}{p(i)} z^\tau$
- $\tau = \tau + 1$
- end

end

We shall analyse in what follows the properties of the algorithm GDPM. Let

$$r = rank(A) \geq 1 \text{ and } 1 = i_1 < i_2 < \dots i_r \leq m \quad (25)$$

indices of a set of linearly independent rows $A_{i_1}, \dots, A_{i_r}$, i.e.

for $i \in [i_1+1, i_2-1]$, A_i depends linearly on A_{i_1}

for $i \in [i_2+1, i_3-1]$, A_i depends linearly on A_{i_1}, A_{i_2}

. . . .

for $i \in [i_{r-1}+1, i_r-1]$, A_i depends linearly on $A_{i_1}, \dots A_{i_{r-1}}$

for $i \in [i_r+1, m]$, A_i depends linearly on $A_{i_1}, \dots A_{i_r}$

We shall use in what follows the notation $z_q^{(k)}$ for the vectors generated during the algorithm GNS (as in NS, section 2). The "evolution" of these vectors during the algorithm is as follows.

for $i \in [i_1+1, i_2-1]$, $\{z_2^{(1)}, \dots, z_n^{(1)}\} \perp A_{i_1} = A_1$

for $i \in [i_2, i_3-1]$, $\{z_3^{(2)}, \dots, z_n^{(2)}\} \perp A_{i_1}, A_{i_2}$

. . . .

for $i \in [i_{r-1}, i_r-1]$, $\{z_r^{(r-1)}, \dots, z_n^{(r-1)}\} \perp A_{i_1}, A_{i_2}, \dots, A_{i_{r-1}}$

for $i \in [i_r, m]$, $\{z_{r+1}^{(r)}, \dots, z_n^{(r)}\} \perp A_{i_1}, A_{i_2}, \dots, A_{i_r} \Leftrightarrow$

$$\{z_{r+1}^{(r)}, \dots, z_n^{(r)}\} \perp A_1, A_2, \dots, A_m \Leftrightarrow \{z_{r+1}^{(r)}, \dots, z_n^{(r)}\} \subset \mathcal{N}(A) \quad (26)$$

Moreover, it is clear from the algorithm GNS that

$$p(i) \neq 0 \;\Leftrightarrow\; i \in \{i_1, i_2, \dots, i_r\} \quad (27)$$

and

$$z_{k+1}^{(k)} \perp A_{i_1}, \dots, A_{i_k},\; k = 1, \dots, r-1. \quad (28)$$

From all the above considerations we derive the following result for GNS.

Proposition 2. *(i) The vectors* $\{z_{r+1}^{(r)}, \dots, z_n^{(r)}\}$ *form a basis in* $\mathcal{N}(A)$*.*
(ii) In the steps 3 and 4 of GNS we obtain a solution of (23) (which generally differs from x_{LS}*.)*

Proof. (i) From (25) we get that $dim(\mathcal{N}(A)) = n - r$. As in [3] we obtain that, during the construction from the algorithm GNS, the vectors $z_{k+1}^k, \dots, z_n^k$ are linearly independent, $\forall k = 1, \dots, r$. In particular, the $n-r$ vectors in the last obtained set $\{z_{r+1}^{(r)}, \dots, z_n^{(r)}\}$ will form a basis $\mathcal{N}(A)$.
(ii) According to (25) the consistent problem (23) is equivalent with

$$\langle A_{i_k}, x\rangle = b_{i_k}, \quad \forall k = 1, \dots, r. \quad (29)$$

Moreover, from all the above notations and considerations, it results that Steps 3 and 4 in the algorithm GNS can be (formally) written as

Step 3-1 (Initialization): any vector $x \in H_1$

Step 4-1 for k=2:r

$i = i_k$

$x = P_{H_i}^{z_k^{k-1}}(x) = x - \frac{\langle x, A_i \rangle - b_i}{p(i)} z_k^{k-1}$

end

end

By using (29), the definition of H_i in (6), and the arguments in [3] page 1162, we obtain that the resulting vector x in the Step 4-1 will satisfy

$$x \in H_{i_k}, \ \forall k = 1, \dots, r, \tag{30}$$

i.e., according to (29) $x \in S(A; b)$. □

Remark 4. *According to the equivalent consistent formulations (3)-(4), we can directly apply the GNS algorithm to them. But, an unpleasant aspect related to the application of the GNS algorithm to (3) consists on the presence of the product $A^T A$ as the system matrix. A row in this matrix is of the form $A_i^T A$ which determines a (too) big computational effort in the steps 2 - 4 of the algorithm GNS.*
According to the second equivalent formulation, if we denote by B the system matrix from (4), its row B_i is given by

$$B_i = \begin{cases} \left[e_i^T, A_i^T \right], & 1 \leq i \leq m \\ \left[(A^{i-m})^T, 0 \right], & m+1 \leq i \leq m+n \end{cases} \tag{31}$$

This tell us that the computational effort in steps 2-4 of GNS are comparable to NS.

A detailed analysis of these aspects, together with an efficient implementation of GNS, and systematic numerical experiments and comparisons with other algorithms, will be the subject of a future paper.

Final comments. Other considerations essentially related to the computation of a basis for the null space of a given matrix A can be found in papers [1, 5, 6, 9, 10]. They use in this respect QR or SVD decompositions of the matrix A, but they are not concerned with a solution of a given system or least squares problem.

Acknowledgements. The author thanks to the anonymous referees for their valuable comments.

References

[1] M. Arioli, L. Baldini: A backward error analysis of a null space algorithm in sparse quadratic programming, Siam J. Matrix Anal. Appl. , 23(2)(2001), $425 - 442$.

[2] M. Benzi: A direct row-projection method for sparse linear systems. Ph.D. Thesis, Department of Mathematics, North Carolina State University, Raleigh, NC, 1993.

[3] M. Benzi M., C. Mayer: A direct projection method for sparse linear systems, SIAM J. Sci. Comput., 16(5)(1995), $1159 - 1176$.

[4] M. Benzi, C. Mayer, M. Tuma: A sparse approximate inverse preconditioner for the conjugate gradient method, SIAM J. Sci. Comput., 17(5)(1996), $1135 - 1149$.

[5] Anthony J. Cox, Nicholas J. Higham: Accuracy and stability of the null space method for solving the equality constrained least squares problem, BIT, vol. 439(1)(1999)4, $34 - 50$.

[6] X. Liu, Y. Yuan: A null-space primal-dual interior-point algorithm for nonlinear optimization with nice convergence properties, Math. Program. , Ser. A 125(2010), $163 - 193$.

[7] A. Nicola, C. Popa: Kaczmarz Extended versus augmented system solution in image reconstruction, ROMAI Journal, 6(1)(2010), $145 - 152$.

[8] C. Popa: Projection algorithms - classical results and developments. Applications to image reconstruction, Lambert Academic Publishing - AV Akademikerverlag GmbH & Co. KG, Saarbrücken, Germany, 2012.

[9] B. Recht, W. Xu, B. Hassibi: Null space conditions and thresholds for rank minimization, Math. Program, Ser. B 127(2011), 175202.

[10] A. Sharma et al: Null space based feature selection method for gene expression data, Int. J. Mach. Learn. & Cyber., 3(2012), 269276.