

Group Lapped Iterative Technique for Fast Solution of Large Linear Systems

José M. Bioucas Dias and José M. Nunes Leitão

Instituto Superior Técnico

Departamento de Engenharia Electrotécnica e de Computadores and

Instituto de Telecomunicações

1096 Lisboa Codex, PORTUGAL.

Email:bioucas@lx.it.pt Phone: +351 1 8418466 Fax: +351 1 8418472

Abstract

This work presents an iterative method for solving systems of linear equations. The technique is very much in the spirit of *group iterative methods*, allowing, however, *lapped groups*. For a wide class of matrices, the *group lapped (GL) technique*, herein introduced, improves the convergence rate by a large factor, demanding approximately the same number of operations per iteration as *group iterative methods*. For systems in which any subsystem is nonsingular and whose inverse matrix elements are null above (below) some upper (lower) diagonal, the exact solution is reached in just one step. Features of the GL method are illustrated with a numerical example

I. INTRODUCTION

The need for solving symmetric positive definite (SPD) linear systems $Ax = b$ (with $x \in \mathfrak{R}^N$ and $A = A^T$) arises often in signal and image processing applications. The methods for finding $x^* = A^{-1}b$ can be classified as *direct* and *iterative* [1]. For large systems, iterative methods are preferred to direct methods [1]; despite the infinite time they generally need to find x^* , they often yield a solution within an acceptable error with fewer operations than direct methods. Moreover, round off errors¹ (or any other error) are dumped out as the process evolves [1, 2, 3].

The Jacobi (J), the Gauss-Seidel (GS), and the successive overrelaxation (SOR) methods are, probably, the most known and widely used iterative techniques. They belong to the class of *linear stationary*

¹For large and/or ill-conditioned systems, rounding errors due to floating-point arithmetic are, frequently, the main problem of direct methods. Rounding errors can severely degrade the solutions found.

iterative methods of first degree [2]. They are also classified as *point iterative*, since each iteration can be implemented by solving simple equations for each system component.

Group iterative methods [2] resemble the point iterative ones, replacing each individual component by a group, such that each component belongs to one and only one group [2]. If the groups form a *partition* of the set $S = \{1, \dots, N\}$, the resulting method is known as a *block method* [3]. Each of the above referred point methods has a correspondent block method; namely, the block Jacobi (BJ), the block Gauss-Seidel (BGS), and the block successive overrelaxation (BSOR).

Block iterative methods were developed with the purpose of increasing the convergence rate of the respective point methods. Assuming that A is a M -matrix, the BJ and BGS converge at least as rapidly as the respective *point* counterparts [2]. On the other hand, if A is *Stieltjes* and π -consistently ordered, then the BSOR method, implemented with the optimum relaxation factor, converges faster than the SOR [2]. It should be stressed that determining the exact or approximate relaxation factor, necessary for the SOR and the BSOR methods, frequently has such a high cost that the method is impracticable².

The main shortcoming of block methods is that the error, after each iteration, tends to be larger on the block boundaries than on its interior (this is illustrated in section IV). The *group lapped (GL) iterative method*, presented in this work, operates (as group methods) on groups of components. However, contrarily to the block methods, in the GL scheme groups are not disjoint. By overlapping the groups in a proper manner, the distance between the components being updated and the ones already updated is kept constant. This is crucial concerning the achievement of greater convergence rates. On the other hand,

²A remarkable exception occurs whenever A has the so-called property A [2].

approximately, the same computational burden as the Gauss-Seidel method.

II. GROUP LAPPED METHOD

The GL method is supported on the iteration groups³, in which the set of indices $S = \{1, \dots, N\}$ is divided into ordered subsets S_i (non necessarily disjoint), such that $\cup_i S_i = S$; formally:

Definition 1: An *ordered segmentation* g of $S = \{1, \dots, N\}$ is an ordered collection of subsets $S_i \subset S$, with $i = 1, \dots, n_g$ such that $\cup_i S_i = S$. Two ordered segmentations g and g' given by S_1, \dots, S_{n_g} and $S'_1, \dots, S'_{n_{g'}}$ are identical if $n_g = n_{g'}$ and if $S_1 = S'_1, \dots, S_{n_g} = S'_{n_g}$.

Definition 1 differs from an *ordered grouping* [2] in that the sets S_i in the latter are disjoint.

Here, we restrict the GL method to the one-dimensional (1D) oriented segmentation

$g_1(D) : S_i = \{i, i+1, \dots, i+D-1\} \quad i = 1, \dots, N-D+1$, where

which is adequate whenever A has all its significant elements *close* to the principal diagonal. This is usually the picture in most 1D processing schemes. For 2D problems defined on regular lattices, a segmentation based on a group of overlapped lines or columns is a better choice [4].

We now introduce a set of matrices depending on A and on the segmentation g :

Definition 2: Given an ordered segmentation g , define the diagonal matrix $D_i = \text{diag}(I_{S_i}(k), k = 1, \dots, N)$, where I_{S_i} is the indicator function of set S_i . Define also $\overline{D}_i = I - D_i$. Given A , let Q_i be a matrix obtained from A by setting to zero all rows r and columns c such that $r \notin S_i$ and $c \notin S_i$. Matrix \overline{Q}_i is obtained from A by setting to zero all rows r such that $r \notin S_i$, and all columns c such that $c \in S_i$. Further, let $G_i = Q_i + \overline{D}_i$ and $H_i = -\overline{Q}_i + \overline{D}_i$.

We now formally introduce the GL method:

GL Method *Take the linear system $Ax = b$, the ordered segmentation g , the cyclic schedule $i = \text{tri}(n, n_g) \equiv (n-1) \bmod n_g + 1$ with $n = 1, 2, \dots$, and the starting vector $x(0)$. Suppose that matrices G_i , for $i = 1, \dots, n_g$ are nonsingular. Then, the GL method generates the sequence $\{x(n)\}$ according to*

$$x(n) = (G_i^{-1} H_i)x(n-1) + G_i^{-1} D_i b \quad n = 1, 2, \dots \quad (1)$$

The following observations are in order:

³The designation of group is not to be understood in the usual mathematical sense.

is fulfilled if A is positive definite. However, the GL method applies to a wider class of matrices.

- The cyclic schedule $i = \text{tri}(n, n_g)$ is not a constraint, since any other schedule can be set by redefining the segmentation g .
- For SPD systems, the GL method seeks the minimum of $F(x(n)) = (1/2)x^T A x - b^T x$ over the set $E(i, n) = \{x \in \mathfrak{R}^N \mid x_j(n) = x_j(n-1); j \notin S_i\}$. Thus, the sequence $F(x(n))$ is monotonically decreasing.
- The i -th update of $x_i(n)$ is an explicit function of b_i, \dots, b_{i+D-1} . In this way the GL method embodies aspects of multigrid philosophy [5].

Successively applying iteration (1), with $i = \text{tri}(n, n_g) = 1, \dots, n_g$, and $n = t n_g + 1, \dots, (t+1)n_g$, one obtains

$$x((t+1)n_g) = M x(t n_g) + N b, \quad (2)$$

$$M = \prod_{i=n_g}^{i=1} G_i^{-1} H_i, \quad (3)$$

$$N = \left(\sum_{i=1}^{n_g-1} \prod_{k=n_g}^{k=i+1} (G_k^{-1} H_k) G_i^{-1} D_i \right) + G_{n_g}^{-1} D_{n_g} b \quad (4)$$

We introduce the subsequence $x(n = t n_g)$, with $t \in \mathcal{N}$ (the set of naturals), which is the output of GL iteration after t full sweeps of index i in (1). The index t in $x(t)$ and the index n in $x(n)$ will distinguish both sequences. Using this notation, equation (2) becomes

$$x(t) = M x(t-1) + N b \quad t = 1, 2, \dots, \quad (5)$$

which is a *linear stationary iterative method of first degree* [2]. If N^{-1} exists, the sequence $\{x(t)\}$ is generated by the splitting $A = B - C$ with $B = N^{-1}$ and $C = N^{-1} M$.

Given an arbitrary initial starting vector $x(0)$, the sequence $x(t)$ generated by (5) is convergent if and only if M is convergent [2] (the spectral radius verifies $\rho(M) < 1$). Of course, the only interesting case is $\lim_{t \rightarrow \infty} x(t) = x^* = A^{-1} b$, which is verified for each of the following cases [4]:

1. under any segmentation, if A is symmetric and positive defined
2. under segmentation $g_1(D)$, if any subdeterminant of A , formed by deleting m rows and the corresponding columns, is non-null, where $0 \leq m < N$, and the elements of $A^{-1} = [r_{ij}]$ meet

the exact solution is reached in just one iteration ($M = 0$).

Consider that $r_{ij} \neq 0$ for $j - i \geq D$. Then, $\rho(M) \neq 0$ and $x(t)$ converges geometrically to x^* . On the other hand, under segmentation $g_1(D)$, the component $x_i(n)$ depends on b_i, \dots, b_{i+D-1} . Thus, the i -th update sees D components b_j towards the future (index j increasing). Contrarily, in the block methods the k -th component inside a block of dimension D sees $D - k$ components b_j towards the future. Accordingly, it is expectable that the error near the block boundaries be greater than the error in the middle of the block. This is a major shortcoming of the block methods.

III. A GAUSS-SEIDEL ITERATION POINT OF VIEW

Consider the i -th iteration (1) with $i + D \leq N$: although the groups S_i have dimension D , only x_i needs to be computed. Notice that: 1) the components x_j , $j = i + 1, \dots, i + D - 1$ will be updated in the next iteration; 2) the new values of the variables in block S_{i+1} , depend on $x_j(t)$ for $j = i + D + 1, \dots, N$, and on $x_j(t + 1)$, for $j = 1, \dots, i$. Thus, in the i -th iteration, only $x_i(t + 1)$ needs to be updated, being wasteful to do the complete block computation⁴. With the above facts in mind, one concludes that the GL output (1) at $n = tn_g$ is given by

$$x_i(t+1) = \left(b'_i - \sum_{k=1}^{k=i-1} a'_{ik} x_k(t+1) - \sum_{k=i+1}^{k=N} a'_{ik} x_k(t) \right), \quad (6)$$

for $i = 1, \dots, N$, where

$$\begin{cases} b'_i &= [G_j^{-1} D_j b]_{i1} \\ a'_{ik} &= [G_j^{-1} H_j]_{ik}, \end{cases} \quad (7)$$

and

$$j = \begin{cases} i & i \leq N - D \\ (N - D + 1) & i > N - D. \end{cases} \quad (8)$$

Equation (6) defines a GS iteration over the system $A'x = b'$ with $A' = [a'_{ij}]$, $b' = [b'_i]^T$, and

$$A' = \begin{bmatrix} 1 & 0 & \cdots & 0 & * & \cdots & \cdots & * \\ * & 1 & 0 & \cdots & 0 & * & \cdots & * \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ * & \cdots & * & 1 & 0 & \cdots & 0 & * \\ * & \cdots & \cdots & * & \boxed{\begin{matrix} I_D \end{matrix}} & \cdots & \cdots & * \\ \vdots & \cdots & \cdots & \vdots & & & & \\ * & \cdots & \cdots & * & & & & \end{bmatrix}, \quad (9)$$

⁴Except for the last block $i = N - D + 1$, where all components have to be computed.

the splitting $A' = G' - H'$, where G' is the lower triangular part of A' , the GS sequence (6) is also given by

$$G'x(t+1) = H'x(t) + b'. \quad (10)$$

As D increases, A' tends to a lower triangular and H' to the null matrix. If H' is a null matrix, then, the solution will be found in just one step; each element $x_i(1)$ is recursively determined from $x_j(1)$ for $j = 1, \dots, i - 1$. However, if $H' \neq 0$, sequence (10), can be thought of as a balance between *recursiveness* and *iterativeness*.

A thorough analysis of equations (8) leads to the conclusion that $A' = TA$ and $b' = Tb$, with $T = [t_{ij}]$ being an upper triangular matrix depending on matrices G_i^{-1} with elements $t_{ij} = 0$ if $j - i \geq D$. This confers to matrix A' the interesting property of having the same number of non-null diagonals as matrix A .

As D increases, the eigenvalues of A' tend to be closer to one (in magnitude) than those of A . Thus, the GL method can be understood as a *preconditioning* technique [6]. This technique is applied to problems exhibiting slow convergence. For sparse matrices, preconditioning has normally the disadvantage of increasing the number of non-null elements. Therefore, the reduction in the number of iterations must compensate the extra computations per iteration. Given that the GL method maintains the same number of non-null diagonals, convergence is speeded up without increasing the computational effort.

A parallel algorithm embodying the GL spirit, and a comparison with *multisplitting* methods [7] is proposed in [4].

IV. NUMERICAL RESULTS

The Toeplitz matrix $A = [a_{\tau=j-i}]$ with $a_\tau = \exp\left[-\left(\frac{\tau}{a}\right)^2\right]$ is considered. Results in Table 1 were computed for $a = \sqrt{3}$ and $N = 64$. Symbols ρ_{GL} , ρ_{BGS} , and ρ_{SOR} denote the spectral radii of GL, BGS, and SOR iteration matrices, respectively. The convergence rate $R(\cdot)$ is given by $R(\cdot) = -\ln_{10}(\rho(\cdot))$. The *condition number* (given by $\kappa(A) = \sigma_n(A)/\sigma_1(A)$, where $\sigma_n(A)$ and $\sigma_1(A)$ are the largest and the smallest singular values of A , respectively) of matrix A , for $a = \sqrt{3}$, is $\kappa(A) \simeq 800$. The SOR spectral radius is $\rho_{SOR} = 0.93666$, leading to the ratio $R(SOR)/R(GS) = 8.4$.

The elements of $A^{-1} = [r_{ij}]$, verify $|r_{ij}/r_{ii}| \leq 0.1$, for $|\tau| = |j - i| > 10$. Thus, the high convergence rate of GL method for $D = 10$ ($R(GL) = 2.28$) is essentially in accordance with the result referred in Section II, concerning one-sided banded A^{-1} matrices.

The superior performance of the GL method,

D	ρ_{GL}	$\overline{R}(GS)$	ρ_{BGS}	$\overline{R}(GS)$
GS \equiv 1	0.99227	1	—	—
2	0.95354	6.13	0.97307	3.52
3	0.85930	19.54	—	—
4	0.71047	44.06	0.95525	5.90
5	0.53687	80.15	—	—
10	0.05264	379.42	0.92107	10.60

Table 1: Spectral radii and convergence ratios.

compared with BGS, is evident for $D \geq 1$. Compared with the SOR method, the GL has much higher convergence rate for $D \geq 3$.

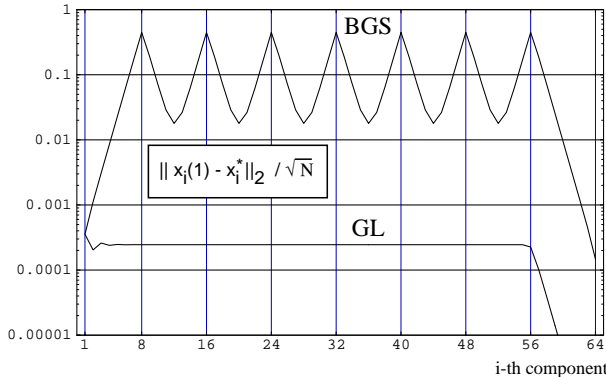


Figure 1: Error components for GL and BGS methods, after the first iteration.

Assume now that $a = 1$. Although smaller, spectral radii ρ_{GL} , ρ_{BGL} , and ρ_{SOR} exhibit the same behavior evidenced for $a = \sqrt{3}$. Fig. 1 plots the error components $\|x_i(1) - x_i^*\|$, for $i = 1, \dots, 64$, with $b = 0$, $x(0) = [1, \dots, 1]^T$, and $D = 8$, after the first iteration, for the BGS and the GL methods, respectively. The BGS method displays an error as large as 600 times the GL error. This was to be expected given the large magnitude of $\rho_{BGS}/\rho_{GL} \simeq (0.17)/(2.72 \cdot 10^{-6})$. The interesting aspect is the shape of the BGS error, exhibiting maxima at $i = 8, 16, 24, 32, 40, 48$, and 56, and minima at $i = 1, 12, 20, 28, 36, 44$, and 52. A crude justification⁵ is the following: the error of each variable inside the block increases with the errors of variables outside the block and decreases as the distance to the nearest boundary grows. In contrast, the GL method keeps the distance to the variable with larger error at a constant value of 8; the exception is the last block, this not being a problem (for both methods) if the variables in the last group do not depend on distant groups.

⁵This argument is valid for matrices whose significant inverse matrix elements are gathered around the main diagonal.

In this paper, we proposed an iterative algorithm for solving large systems of linear equations. Although applicable to a wide class of system matrices, it was firstly thought for SPD ones. The i -th and $(i + 1)$ -th iterations minimize the quadratic function $F(x) = (1/2)x^T Ax - b^T x$ with respect to the overlapped groups of variables S_i and S_{i+1} , respectively. Hence the name *group lapped* (GL).

For systems in which a given component depends vanishingly on future (past) data components, the method achieves high convergence rates compared with the point and block methods, with, approximately, the same complexity per iteration of the Gauss-Seidel scheme applied to the original system. If the inverse system matrix is one-sided banded, there exist segmentations (a choice of the iteration groups), such that the exact solution is found in just one iteration.

REFERENCES

- [1] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation. Numerical Methods*, Prentice Hall, New Jersey, 1989.
- [2] D.M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.
- [3] Richard S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englehood Cliffs, NG, 1962.
- [4] J.B. Dias, *Estimation of Spectral Moments: Conceptual and Computational Aspects*, PhD thesis, Dep. Eng. Elec. Comp.- I.S.T., Lisbon, 1994, (In Portuguese).
- [5] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer-Verlag, New York, 1985.
- [6] D.J.Evans, *Preconditioning Methods: Analysis and Applications*, Gordon and Breach, New York, 1983.
- [7] R. E. White, “Multisplittings of a symmetric positive definite matrix”, *SIAM J. Matrix Anal. Appl.*, vol. 11, pp. 69–82, 1990.