

A note on the relation of Gaussian elimination to the conjugate directions algorithm

J. Tenne*

S. W. Armfield*

(Received 5 November 2004, revised 15 September 2005)

Abstract

This work examines the relation between Gaussian elimination and the conjugate directions algorithm [Hestenes and Steifel, 1952]. Analysis is extended to the case where the sequence of the conjugated vectors is modified, which is shown to result in reordering of the solution vector. Based on these analyses an algorithm is described which combines Gaussian elimination with a look-ahead algorithm. The purpose of the algorithm is to employ Gaussian elimination on a system of smaller order and to use this solution to approximate the solution of the original system. The algorithm was tested on a range of linear systems and performed well when the components in the solution vector varied by large magnitude.

*School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, NSW, AUSTRALIA. <mailto:joel.tenne@aeromech.usyd.edu.au>, <mailto:armfield@aeromech.usyd.edu.au>

See <http://anziamj.austms.org.au/V46/CTAC2004/Tenn> for this article, © Austral. Mathematical Soc. 2005. Published October 7, 2005. ISSN 1446-8735

Contents

1	Introduction	C972
2	Background	C973
2.1	Gaussian elimination	C974
2.2	The Gram–Schmidt algorithm	C974
2.3	The conjugate directions algorithm	C975
3	The basic relations	C976
4	Extension to a different sequence of conjugated vectors	C979
5	A look-ahead algorithm for Gaussian elimination	C980
6	Summary	C984
	References	C985

1 Introduction

It is common to solve a set of linear equations using either Gaussian elimination or a variant of the conjugate directions algorithm. Accordingly, it is important to obtain a thorough understanding of their mechanics. It was observed initially in [6] and later in [4, p.173–177] that when the coefficient matrix is symmetric and positive definite Gaussian elimination is derived from a conjugate Gram–Schmidt algorithm and the conjugate directions algorithm. This work proves a relation between Gaussian elimination and the conjugate Gram–Schmidt algorithm and extends the analysis to the case when the sequence of conjugated vectors is changed. Based on these analyses an algorithm is described such that a solution with a sufficiently small residual can be obtained by employing Gaussian elimination on a linear system

of reduced order compared to the original linear system.

Besides the above references the relations between Gaussian elimination and the conjugate directions algorithm were found to be briefly mentioned only in [1, p.463] and [2], presumably since attention focused on the more efficient conjugate gradient variant. However, the present analysis is beneficial since it provides a uniform understanding and relates what appears to be disjoint approaches. Section 2 provides a concise background. Section 3 examines the basic relations. Section 4 extends the results to the case when the sequence of the conjugated directions is changed. Section 5 describes an algorithm which combines Gaussian elimination with the conjugate directions algorithm to produce an approximate solution and discusses computational results.

2 Background

The three topics relevant to this work are Gaussian elimination, the Gram–Schmidt algorithm and the conjugate directions algorithm. Attention is given only to aspects relevant to the present work. Full details of these algorithms can found in many good references, such as [3, 7].

The following standard notation is used in this paper. A matrix is given in bold uppercase (for example, \mathbf{M}), its i th row vector is \mathbf{M}_i and its element at the i th row and the j th column is $\mathbf{M}_{i,j}$. Vectors are given in bold, for example, \mathbf{v} . A vector set is denoted by calligraphic uppercase letters, for example, $\mathcal{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n\}$ denotes that \mathcal{Q} is a vector set containing $\mathbf{q}_1, \dots, \mathbf{q}_n$. The i th component of a vector \mathbf{v} is given by v_i and it is not bold to distinguish it from a vector within a set. Scalars are not emphasized and a superscript k denotes the k th update of the variable (for example, $\mathbf{M}^{(k)}$). The standard Euclidean inner product of two vectors is $\langle \mathbf{u}, \mathbf{v} \rangle$ and the identity matrix is $\mathbf{I} \in \mathbb{R}^{n \times n}$. The order of a matrix is defined explicitly except where it is inferred from the context.

2.1 Gaussian elimination

Gaussian elimination solves the linear equation set problem

$$\mathbf{Ax} = \mathbf{b}, \quad \mathbf{A} \in \mathbb{R}^{n \times n}, \quad \mathbf{b} \in \mathbb{R}^n, \quad (1)$$

by the factorization $\mathbf{A} = \mathbf{LU}$ where \mathbf{L} is lower triangular and \mathbf{U} is upper triangular. The solution vector is found by solving two triangular systems such that $\mathbf{Ly} = \mathbf{b}$ and $\mathbf{Ux} = \mathbf{y}$. The factorization is conveniently expressed as premultiplication by lower triangular matrices which describe the linear operations on the rows of \mathbf{A} . Accordingly at a stage $k > 1$ of the factorization the coefficient matrix $\mathbf{A}^{(k)}$ is

$$\mathbf{L}_k \cdots \mathbf{L}_2 \mathbf{A} = \mathbf{A}^{(k)} \quad (2)$$

and

$$\mathbf{L}_{n+1} \cdots \mathbf{L}_2 \mathbf{A} = \mathbf{U}. \quad (3)$$

When the coefficient matrix is symmetric and positive definite Gaussian elimination is stable so pivoting is unnecessary and hence the factorization is unique.

2.2 The Gram–Schmidt algorithm

Given a linearly independent vector set $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ the algorithm generates a mutually orthogonal vector set \mathcal{Q} with the same span using successive orthogonal projections, that is,

$$\mathcal{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_n\}, \quad \langle \mathbf{q}_i, \mathbf{q}_j \rangle = 0 \text{ for all } i \neq j, \quad \text{span}\{\mathcal{V}\} = \text{span}\{\mathcal{Q}\}, \quad (4)$$

$$\mathbf{q}_1 = \mathbf{v}_1, \quad \mathbf{q}_i = \mathbf{v}_i - \sum_{j=1}^{i-1} \frac{\langle \mathbf{v}_i, \mathbf{q}_j \rangle}{\langle \mathbf{q}_j, \mathbf{q}_j \rangle} \mathbf{q}_j = \left(\mathbf{I} - \sum_{j=1}^{i-1} \frac{\mathbf{q}_j \mathbf{q}_j^T}{\langle \mathbf{q}_j, \mathbf{q}_j \rangle} \right) \mathbf{v}_i, \quad i \geq 2. \quad (5)$$

When \mathbf{A} is symmetric and positive definite it is possible to define the \mathbf{A} -norm of a vector as $\|\mathbf{x}\|_{\mathbf{A}} = \sqrt{\langle \mathbf{A}\mathbf{x}, \mathbf{x} \rangle}$ since $\|\mathbf{x}\|_{\mathbf{A}} > 0$ for all $\mathbf{x} \neq \mathbf{0}$ and $\|\mathbf{x}\|_{\mathbf{A}} = 0$ only if $\mathbf{x} = \mathbf{0}$. Accordingly it is possible to extend the orthogonal Gram–Schmidt algorithm so as to generate a vector set \mathcal{Q} which is mutually \mathbf{A} -orthogonal or *conjugate* using \mathbf{A} -orthogonal projections, that is,

$$\mathcal{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_n\}, \quad \langle \mathbf{A}\mathbf{q}_i, \mathbf{q}_j \rangle = 0 \text{ for all } i \neq j, \quad \text{span}\{\mathcal{V}\} = \text{span}\{\mathcal{Q}\}, \quad (6)$$

$$\mathbf{q}_1 = \mathbf{v}_1, \quad \mathbf{q}_i = \mathbf{v}_i - \sum_{j=1}^{i-1} \frac{\langle \mathbf{A}\mathbf{v}_i, \mathbf{q}_j \rangle}{\langle \mathbf{A}\mathbf{q}_j, \mathbf{q}_j \rangle} \mathbf{q}_j = \left(\mathbf{I} - \sum_{j=1}^{i-1} \frac{\mathbf{q}_j \mathbf{q}_j^T \mathbf{A}}{\langle \mathbf{A}\mathbf{q}_j, \mathbf{q}_j \rangle} \right) \mathbf{v}_i, \quad i \geq 2. \quad (7)$$

2.3 The conjugate directions algorithm

When \mathbf{A} is symmetric and positive definite it is possible to solve (1) as an optimization problem based on the level surfaces of positive definite quadratic functions [4, 5]. In this case the solution of the linear equation set problem can be obtained by minimizing the positive definite quadratic function

$$\varphi(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}, \quad \varphi(\mathbf{x}) = \frac{1}{2} \langle \mathbf{A}\mathbf{x}, \mathbf{x} \rangle - \langle \mathbf{b}, \mathbf{x} \rangle, \quad (8)$$

since its gradient is

$$\nabla \varphi(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}, \quad (9)$$

and accordingly the unique minimizer \mathbf{x}^* of $\varphi(\mathbf{x})$ satisfies $\nabla \varphi(\mathbf{x}^*) = \mathbf{0}$ or $\mathbf{A}\mathbf{x}^* = \mathbf{b}$.

The conjugate directions algorithm [4, 6] obtains the minimizer \mathbf{x}^* by successively minimizing $\varphi(\mathbf{x})$ along directions which are mutually conjugated, starting from an initial guess $\mathbf{x}^{(1)}$. Disregarding finite precision considerations, the method obtains the exact minimizer \mathbf{x}^* in exactly n minimizations.

At intermediate stages $k < n + 1$ the algorithm provides an approximate solution $\mathbf{x}^{(k)}$ and its corresponding residual $\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}$ such that

$$\langle \mathbf{x}^{(k)}, \mathbf{r}^{(k)} \rangle = 0. \quad (10)$$

Numerous variants have been derived based on the above basic approach including the popular conjugate gradients method. Algorithm 1 describes the conjugate directions algorithm relevant to the present work.

Algorithm 1: The conjugate directions algorithms.

Require: $\mathcal{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_n\}$, $\text{span}\{\mathcal{Q}\} = \mathbb{R}^n$, $\langle \mathbf{A}\mathbf{q}_i, \mathbf{q}_j \rangle = 0$ for all $i \neq j$, $\mathbf{A} \in \mathbb{R}^{n \times n}$ symmetric positive definite, $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{x}^{(1)} \in \mathbb{R}^n$.

1: $\mathbf{r}^{(1)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(1)}$

2: **for** $i = 1$ to n **do**

3: $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \frac{\langle \mathbf{q}_i, \mathbf{r}^{(i)} \rangle}{\langle \mathbf{A}\mathbf{q}_i, \mathbf{q}_i \rangle} \mathbf{q}_i$

4: $\mathbf{r}^{(i+1)} = \mathbf{r}^{(i)} - \frac{\langle \mathbf{q}_i, \mathbf{r}^{(i)} \rangle}{\langle \mathbf{A}\mathbf{q}_i, \mathbf{q}_i \rangle} \mathbf{A}\mathbf{q}_i$

5: **end for**

6: Returns $\mathbf{x}^{(n+1)} = \mathbf{x}^* : \mathbf{A}\mathbf{x}^* = \mathbf{b}$

3 The basic relations

When Gaussian elimination is employed on a coefficient matrix \mathbf{A} which is symmetric and positive definite the matrix $\mathbf{A}^{(k)}$ may be obtained by a conjugate Gram–Schmidt algorithm. This is shown initially for a single step and then extended to the general case.

Defining $\mathbf{A}^{(1)} = \mathbf{A}$ and employing Gaussian elimination on (1) then after one step the resultant system is

$$\mathbf{A}^{(2)}\mathbf{x} = \mathbf{b}^{(2)}, \quad \mathbf{A}^{(2)} = \begin{pmatrix} a & \mathbf{w}^T \\ \mathbf{0} & \mathbf{B} \end{pmatrix}. \quad (11)$$

The row-wise elimination along the first column of $\mathbf{A}^{(1)}$ is

$$\mathbf{A}_i^{(2)} = \mathbf{A}_i^{(1)} - \frac{\mathbf{A}_{i,1}^{(1)}}{\mathbf{A}_{1,1}^{(1)}} \mathbf{A}_1^{(1)}. \quad (12)$$

Let $\mathcal{Q}^{(1)}$ be the natural coordinates vector set in \mathbb{R}^n so $\mathbf{q}_i^{(1)}$ is all zeros except for the i th component which is one. Accordingly the term

$$\mathbf{A}_{i,1}^{(1)} = \left\langle \mathbf{A}\mathbf{q}_i^{(1)}, \mathbf{q}_1^{(1)} \right\rangle, \quad (13)$$

and (12) is written as

$$\mathbf{A}_i^{(2)} = \mathbf{A} \left(\mathbf{q}_i^{(1)} - \frac{\left\langle \mathbf{A}\mathbf{q}_i^{(1)}, \mathbf{q}_1^{(1)} \right\rangle}{\left\langle \mathbf{A}\mathbf{q}_1^{(1)}, \mathbf{q}_1^{(1)} \right\rangle} \mathbf{q}_1^{(1)} \right) = \mathbf{A}\mathbf{q}_i^{(2)}. \quad (14)$$

The expression in parenthesis is a conjugation of the vectors $\mathbf{q}_2^{(1)}, \dots, \mathbf{q}_n^{(1)}$ to the vector $\mathbf{q}_1^{(1)}$ as described in (7). Let the matrix

$$\mathbf{Q}_i^{(k)} = \mathbf{q}_i^{(k)}, \quad \mathbf{q}_i^{(k)} \in \mathcal{Q}^{(k)}, \quad (15)$$

that is, its i th row vector is the i th vector in the set $\mathcal{Q}^{(k)}$. Relation (14) is written in matrix notation as

$$\mathbf{A}^{(2)} = \left(\mathbf{A}\mathbf{Q}^{(2)} \right)^T = \left(\mathbf{Q}^{(2)} \right)^T \mathbf{A}, \quad (16)$$

hence after one step of Gaussian elimination the coefficient matrix $\mathbf{A}^{(2)}$ is the image of \mathbf{A} on the vector basis $\left(\mathbf{Q}^{(2)} \right)^T$ or alternatively $\left(\mathbf{Q}^{(2)} \right)^T = \mathbf{L}_2$ which is the lower triangular matrix of Gaussian elimination (§2.1).

Also, from (1) and (16),

$$\mathbf{b}^{(2)} = \left(\mathbf{Q}^{(2)} \right)^T \mathbf{b}, \quad (17)$$

and these relations are extended by the following lemma.

Lemma 1 *When Gaussian elimination is employed on a symmetric positive definite coefficient matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ then the matrix at stage k , denoted $\mathbf{A}^{(k)}$, is obtained by a conjugate Gram–Schmidt algorithm using a vector set $\mathcal{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_n\}$ such that \mathbf{q}_i is the i th natural coordinate axis in $\mathbb{R}^{(n-k+1)}$.*

Proof: It is easy to prove that when \mathbf{A} is symmetric and positive definite then the trailing sub-matrix \mathbf{B} in (11) is also symmetric and positive definite, for example by a Cholesky factorization. Since $\mathbf{B} \in \mathbb{R}^{(n-1) \times (n-1)}$ and the next step of Gaussian elimination operates only on \mathbf{B} then relations (11)–(21) are valid if a vector set $\mathcal{Q}^{(2)} = \{\mathbf{q}_1, \dots, \mathbf{q}_{n-1}\}$ such that \mathbf{q}_i is the i th natural coordinate axis in \mathbb{R}^{n-1} and the matrix $\mathbf{Q}^{(2)}$ is such that $\mathbf{Q}_i^{(2)} = \mathbf{q}_i$. By induction it follows that at each step of the Gaussian elimination there exists a symmetric positive definite sub-matrix and the relations hold with a suitable vector set $\mathcal{Q}^{(k)}$. Hence $\mathbf{A}^{(k)}$ is derived from a conjugate Gram–Schmidt algorithm. ♠

By the relation to the conjugated Gram–Schmidt algorithm and since \mathbf{A} is symmetric and positive definite, it is possible to generate during the elimination an approximate solution with an associated residual based on Algorithm 1 and the conjugated vector set $\mathcal{Q}^{(2)}$. If $\mathbf{x}^{(1)} = \mathbf{0}$ from which $\mathbf{r}^{(1)} = \mathbf{b}$ then $\mathbf{b}^{(2)}$ is the sum of two vectors [4, p.174–175]

$$\mathbf{b}^{(2)} = \left(\mathbf{q}_1^{(1)T} \mathbf{b}, 0, \dots, 0 \right) + \mathbf{r}^{(2)}, \quad (18)$$

where $\mathbf{r}^{(2)}$ is the residual and has the form

$$\mathbf{r}^{(2)} = \left(0, r_2^{(2)}, \dots, r_n^{(2)} \right). \quad (19)$$

From the orthogonal relation (10),

$$\langle \mathbf{x}^{(2)}, \mathbf{r}^{(2)} \rangle = 0, \quad (20)$$

hence

$$\mathbf{x}^{(2)} = \left(x_1^{(2)}, 0, \dots, 0 \right) \in \mathbb{R}^n. \quad (21)$$

Relations (11)–(21) were observed to hold using the vector set $\mathcal{Q}^{(k)}$ where $\mathbf{q}_i^{(1)}$ is the i th natural coordinate axis in \mathbb{R}^n and the vectors $\mathbf{q}_i^{(k)}$, $i = 1, \dots, k-1$ are mutually conjugated, that is, for stages $k = 2, \dots, n+1$ and $\mathbf{x}^{(1)} = \mathbf{0}$ the following relations hold with $\mathcal{Q}^{(k)}$ obtained from the relation in (15):

$$\mathcal{Q}^{(k)} : \left\langle \mathbf{A}\mathbf{q}_i^{(k)}, \mathbf{q}_j^{(k)} \right\rangle = 0, \quad 1 \leq i \leq k-1, \quad k-1 < j \leq n, \quad (22)$$

$$\mathbf{A}^{(k)} = \mathcal{Q}^{(k)} \mathbf{A}, \quad (23)$$

$$\mathbf{b}^{(k)} = \mathcal{Q}^{(k)} \mathbf{b}, \quad (24)$$

$$\mathbf{r}^{(k)} = \left(0, \dots, 0, b_k^{(k)}, \dots, b_n^{(k)} \right) \in \mathbb{R}^n, \quad (25)$$

$$\mathbf{x}^{(k)} = \left(x_1^{(k)}, \dots, x_{k-1}^{(k)}, 0, \dots, 0 \right) \in \mathbb{R}^n. \quad (26)$$

4 Extension to a different sequence of conjugated vectors

In the previous section choosing $\mathcal{Q}^{(1)}$ such that $\mathcal{Q}^{(1)} = \mathbf{I}$ defined the sequence of the conjugated vectors. In this section the analysis is extended to the case when the sequence of conjugated vectors is changed. Let $\mathbf{P} \in \mathbb{R}^{n \times n}$ be a non-singular permutation matrix, that is, each row \mathbf{P}_i is all zeros except a single component which is one. In this case the system

$$\mathbf{P}\mathbf{A}\mathbf{P}^T\mathbf{P}\mathbf{x} = \mathbf{P}\mathbf{b} \quad (27)$$

has the same solution as (1) since \mathbf{P} is orthogonal ($\mathbf{P}\mathbf{P}^T = \mathbf{I}$). Furthermore the matrix $\mathbf{P}\mathbf{A}\mathbf{P}^T$ is symmetric since $(\mathbf{P}\mathbf{A}\mathbf{P}^T)^T = \mathbf{P}\mathbf{A}\mathbf{P}^T$ and it is positive definite since \mathbf{P} is non-singular and $\langle \mathbf{P}\mathbf{A}\mathbf{P}^T \mathbf{x}, \mathbf{x} \rangle = \mathbf{x}^T \mathbf{P}\mathbf{A}\mathbf{P}^T \mathbf{x} = \mathbf{y}^T \mathbf{A} \mathbf{y} > 0$ for all $\mathbf{x} \neq \mathbf{0}$.

Equation (27) implies the procedure of Section 3 can be used with the symmetric positive matrix $\bar{\mathbf{A}} = \mathbf{PAP}^T$ and the vector $\bar{\mathbf{b}} = \mathbf{Pb}$. This results in a solution vector whose components are reordered such that $\bar{\mathbf{x}} = \mathbf{Px}$. Since the factorization of $\bar{\mathbf{A}}$ is unique it follows the relations (22)–(26) must hold with $\mathbf{Q}^{(1)} = \mathbf{P}$ to account for the reordering of the components in the solution vector. Thus $\bar{\mathbf{x}}$ can be obtained by either solving (1) with $\mathbf{Q}^{(1)} = \mathbf{P}$ or by initially permuting (1) to obtain (27) and then solving with $\mathbf{Q}^{(1)} = \mathbf{I}$.

5 A look-ahead algorithm for Gaussian elimination

The results of §3 and §4 imply that the vector sequence in $\mathcal{Q}^{(1)}$ determines the sequence of components in the approximate solution vector $\mathbf{x}^{(l)}$, $1 \leq l \leq n+1$ and accordingly in its associated residual. Furthermore, the approximate solution (26) is updated coordinate-wise hence at each stage l it contains at most $l - 1$ non-zero components. Thus the leading $l - 1$ components may be regarded as the solution vector of a linear system of order $l - 1$. These observations motivated the idea that it is possible to reduce the work required to solve (1) when using Gaussian elimination by solving a linear system of smaller order. The solution of the smaller system is used to approximate the solution of the original system. Accordingly the smaller system should be generated so as to minimize the residual of the approximate solution.

To accomplish this, prior to the Gaussian elimination a look-ahead algorithm is employed whose purpose is to generate a linear system from (1) of order $k < n$, referred to as a *reduced* linear system. This system is generated using a permutation matrix \mathbf{P} which is determined by the following procedure.

Let $\mathcal{U}^{(k)}$, $\mathcal{V}^{(k)}$ be the vector sets such that $\mathcal{U}^{(1)}$ is initially empty, that is, $\mathcal{U}^{(1)} = \emptyset$ and $\mathcal{V}^{(1)} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ such that \mathbf{v}_i is the i th natural coordinate

axis in \mathbb{R}^n .

Also let $\mathbf{r}^{(k)}$ be the residual variable such that $\mathbf{r}^{(1)} = \mathbf{b}$. At each stage one vector is removed from $\mathcal{V}^{(k)}$ and one vector is added to $\mathcal{U}^{(k)}$ as follows. Assuming stage $k > 1$ then $\mathcal{V}^{(k)}$ contains $n - k + 1$ vectors and $\mathcal{U}^{(k)} = \{\mathbf{u}_1, \dots, \mathbf{u}_{k-1}\}$. For each vector $\mathbf{v} \in \mathcal{V}^{(k)}$ two vectors are found:

$$\mathbf{u} = \mathbf{u}(\mathbf{v}) = \left(\mathbf{I} - \sum_{j=1}^{k-1} \frac{\mathbf{A}\mathbf{u}_j\mathbf{u}_j^T}{\langle \mathbf{A}\mathbf{u}_j, \mathbf{u}_j \rangle} \right) \mathbf{v}, \quad (28)$$

$$\mathbf{r}(\mathbf{v}) = \mathbf{r}^{(k-1)} - \frac{\langle \mathbf{r}^{(k-1)}, \mathbf{u} \rangle}{\langle \mathbf{A}\mathbf{u}, \mathbf{u} \rangle} \mathbf{A}\mathbf{u}, \quad (29)$$

and a vector $\hat{\mathbf{v}} \in \mathcal{V}^{(k)}$ is chosen such that

$$\hat{\mathbf{v}} : \min \{ \|\mathbf{r}(\mathbf{v})\|_2 \} \text{ for all } \mathbf{v} \in \mathcal{V}^{(k)}, \quad (30)$$

hence $\hat{\mathbf{v}}$ minimizes the residual norm out of all vector in $\mathcal{V}^{(k)}$. Next $\mathbf{u}(\hat{\mathbf{v}})$ is added to $\mathcal{U}^{(k)}$ such that $\mathbf{u}_k = \mathbf{u}(\hat{\mathbf{v}})$, $\hat{\mathbf{v}}$ is removed from $\mathcal{V}^{(k)}$ and $\mathbf{r}^{(k)} = \mathbf{r}(\hat{\mathbf{v}})$. The procedure continues until $\|\mathbf{r}^{(k)}\| \leq \epsilon$, where ϵ is chosen by the user.

The procedure identifies the sequence of vectors which minimizes the residual most quickly. This sequence defines the sequence of components in the solution vector and accordingly it defines a permutation matrix $\mathbf{P} \in \mathbb{R}^{k \times n}$ and a reordered system as in (27) whose solution is

$$\bar{\mathbf{x}} = (\bar{x}_1, \dots, \bar{x}_k) \in \mathbb{R}^k, \quad (31)$$

and the approximate (permuted) solution to the original problem (1) is

$$\mathbf{x}_{\text{app}} = (\bar{x}_1, \dots, \bar{x}_k, 0, \dots, 0) \in \mathbb{R}^n. \quad (32)$$

Three computational issues are considered. First, in order to efficiently generate the conjugated vectors in (28) the coefficient matrix \mathbf{A} should be sparse since the procedure involves matrix-vector multiplications. Second,

in the approximate solution (32) the final $(n - k)$ components are taken as zero and in order for the residual $\mathbf{r}_{\text{app}} = \mathbf{b} - \mathbf{A}\mathbf{x}_{\text{app}}$ to be small these components should have little or no affect on the solution of (1). If it is a priori known which components can be ignored, for example, due to their expected magnitude or the structure of \mathbf{A} , it is trivial to generate a reduced system. Otherwise a reasonable assumption is that the order of the elements of \mathbf{A} is uniform, for example $\mathcal{O}(\mathbf{A}_{i,j}) = 1$ for all $1 \leq i, j \leq n$, and if all components in the solution vector are equally important then the above approximation is valid if some components are much larger (and hence more significant) than others. Accordingly if $\mathbf{x} \in \mathbb{R}^n$ then there are $n \cdot Dx$ significant components where $Dx \in [0, 1]$ represents the density of \mathbf{x} . The insignificant components are smaller by at least a scale factor SF from the significant ones. Thus it is possible to reorder \mathbf{x} such that

$$|x_i| \geq \text{SF} \cdot |x_j| \quad 1 \leq i \leq k, \quad k + 1 \leq j \leq n. \quad (33)$$

For example, this occurs when the solution vector is an update of variables when only variables in a subset are changed. Finally, (28)–(29) can be simplified if the vectors in $\mathcal{V}^{(k)}$ are conjugated to $\hat{\mathbf{v}}$ at each stage.

Algorithm 2 describes the look-ahead algorithm. It was implemented and tested with Gaussian elimination on systems of order $n = 25$ and 50 . Test cases were designed to examine the effect of the density of \mathbf{A} defined by $DA \in [0, 1]$ as percent of non-zero elements in A , the order ratio (SF) and the density of \mathbf{x} (Dx). Each test case examined one combination from the following preset values: (a) $DA = 0.10, 0.25$ (b) $\text{SF} = 100, 1000$ (c) $Dx = 0.25, 0.50$. The elements of \mathbf{A} and \mathbf{x} were generated in random using a uniform distribution and $n \cdot (1 - Dx)$ components in \mathbf{x} were divided by SF to reduce their order. The final values of \mathbf{A} and \mathbf{x} defined the right hand side vector \mathbf{b} and the algorithm was applied as described earlier.

At each test case two variables were recorded: the ratio between the initial to the final residual (r) and the order of the reduced linear system (k). These variables define the efficiency of the procedure since r indicates the

Algorithm 2: The look-ahead algorithm for Gaussian elimination.

Require: $\mathbf{A} \in \mathbb{R}^{n \times n}$ symmetric positive definite, $\mathbf{b} \in \mathbb{R}^n$, $\epsilon_r > 0$.

```

1:  $k = 0$ 
2:  $\mathbf{r}^{(1)} = \mathbf{b}$ 
3:  $\mathcal{U}^{(1)} = \emptyset$ 
4:  $\mathcal{V}^{(1)} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\} : \mathbf{v}_i$  is the  $i$ th natural coordinate vector in  $\mathbb{R}^n$ 
5: repeat
6:    $k = k + 1$ 
7:   for all  $\mathbf{v} \in \mathcal{V}^{(k)}$  do
8:     find  $\mathbf{u}(\mathbf{v})$ ,  $\mathbf{r}^{(k)}(\mathbf{v})$ 
9:   end for
10:   $\hat{\mathbf{v}} : \min \{\|\mathbf{r}(\mathbf{v})\|_2\}$  for all  $\mathbf{v} \in \mathcal{V}^{(k)}$ 
11:   $\mathbf{r}^{(k)} = \mathbf{r}(\hat{\mathbf{v}})$ 
12:   $\mathcal{U}^{(k+1)} = \mathcal{U}^{(k)} \cup \mathbf{u}(\hat{\mathbf{v}})$ 
13:   $\mathcal{V}^{(k+1)} = \mathcal{V}^{(k)} \setminus \hat{\mathbf{v}}$ 
14: until  $\|\mathbf{r}^{(k)}\| \leq \epsilon_r$ 
15: generate  $\mathbf{P}$ 

```

adequacy of the approximate solution and k determines the amount of work required by the Gaussian elimination. These values indicate the performance of the algorithm after convergence, that is, after the approximate residual is considered sufficiently small. Each case was repeated for ten times and the averaged results are summarized in Table 1. The following relations were observed:

- In all cases the algorithm obtained a residual which was $\mathcal{O}(10^{-4})$ of the initial residual using a reduced linear system. The order of the reduced system was affected by a combination of all parameters.
- The parameter SF had the greatest effect on the order of the reduced system. When SF = 100 was used the order of the smaller system was

very close to the full system, but with $SF = 1000$ the smaller system was about half size of the original (depending on Dx).

- Changes in DA and Dx had a minor effect on the order of the reduced system. For example, changing DA from 0.1 (Test 10) to 0.25 (Test 14) increased the order of the reduced system (k) by six from 22 to 28. Doubling Dx from 0.25 (Test 10) to 0.5 (Test 12) increased k by seven from 22 to 29.

Accordingly the algorithm results in a satisfactory residual when components of the solution vector differ a by large magnitude. The density of the coefficient matrix and of the solution vector have a smaller effect on the order of the reduced system.

6 Summary

The relation between Gaussian elimination and the conjugate directions algorithm was analyzed. The relation was extended to the case where the sequence of the conjugated vectors is modified, which was shown to result in reordering of the solution vector. Based on these analyses an algorithm was described which combines Gaussian elimination with a look-ahead algorithm. The purpose of the algorithm is to generate a linear system of reduced order whose solution would approximate the exact solution of the full system. The algorithm was tested on a range of linear systems. It generated satisfactory approximate solutions using significantly smaller linear system when the components in the solution vector varied by a large magnitude.

TABLE 1: Test results with the look-ahead Gaussian elimination algorithm.

Case	n	DA	Dx	SF	r	k
1	25	0.10	0.25	100	6.14e-4	22
2	25	0.10	0.25	1000	9.36e-4	10
3	25	0.10	0.50	100	7.11e-4	22
4	25	0.10	0.50	1000	7.26e-4	13
5	25	0.25	0.25	100	3.71e-4	23
6	25	0.25	0.25	1000	8.28e-4	14
7	25	0.25	0.50	100	5.40e-4	22
8	25	0.25	0.50	1000	8.22e-4	14
9	50	0.10	0.25	100	8.06e-4	44
10	50	0.10	0.25	1000	9.56e-4	22
11	50	0.10	0.50	100	7.21e-4	45
12	50	0.10	0.50	1000	7.90e-4	29
13	50	0.25	0.25	100	3.22e-4	48
14	50	0.25	0.25	1000	7.56e-4	28
15	50	0.25	0.50	100	3.58e-4	48
16	50	0.25	0.50	1000	6.78e-4	31

References

- [1] Owe Axelsson. *Iterative Solution Methods*. Cambridge University Press, Cambridge ; New York, 1994. C973
- [2] Gene H. Golub and Dianne P. O’Leary. Some history of the conjugate gradient and Lanczos algorithms: 1948–1976. *SIAM Review*, 31(1):50–102, 1989. C973
- [3] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore and London, third edition, 1996. C973

- [4] Magnus R. Hestenes. *Conjugate Direction Methods in Optimization*. Number 12 in Applications of Mathematics. Springer-Verlag, New York; Heidelberg; Berlin, 1980. C972, C975, C978
- [5] Magnus R. Hestenes. Conjugacy and gradients in variational theory and analysis. In *Proceedings of the ACM Conference on History of Scientific and Numeric Computation*, pages 71–90, New York, NY, USA, May 1987. ACM, ACM Press. <http://doi.acm.org/10.1145/41579.41586> C975
- [6] Magnus R. Hestenes and Eduard Steifel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952. C972, C975
- [7] Llyod N. Trefethen and David Bau, III. *Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997. C973