*Journal of Prime Research in Mathematics Vol.* **11***(2015), 106-122*

# $A_{19}/B_6$: A NEW LANCZOS-TYPE ALGORITHM AND ITS IMPLEMENTATION

ZAKIR ULLAH[1], MUHAMMAD FAROOQ[2] AND ABDELLAH SALHI[3]

Abstract. Lanczos-type algorithms are mostly derived using recurrence relationships between formal orthogonal polynomials. Various recurrence relations between these polynomials can be used for this purpose. In this paper, we discuss recurrence relations $A_{19}$ and $B_6$ for the choice $U_i(x) = P_i^{(1)}(x)$, where $U_i$ is an auxiliary family of polynomials of exact degree $i$. This leads to new Lanczos-type algorithm $A_{19}/B_6$ that shows superior stability when compared to existing algorithms of the same type. This new algorithm is derived and described here. Computational results obtained with it are compared to those of the most robust algorithms of this type namely $A_{12}$, $A_{12}^{new}$ $A_5/B_{10}$ and $A_8/B_{10}$ on the same test problems. These results are included.

*Key words* : Lanczos algorithm; Systems of Linear Equations; Formal Orthogonal Polynomials
*AMS SUBJECT* : Primary 65F10.

## 1. Introduction

In 1950, the Lanczos algorithm, [26, 13], has been introduced to calculate the eigenvalues of a matrix. However, it has later been adapted for the solution of systems of linear equations (SLEs) where it is now a well established solver. The Lanczos method is an iterative process which, in exact arithmetic, gives the exact solution in at most $n$ number

---

[1] Department of Mathematics, University of Peshawar, Khyber Pakhtunkhwa, 25120, Pakistan. Email: zakirmath728@gmail.com
[2] Department of Mathematics, University of Peshawar, Khyber Pakhtunkhwa, 25120, Pakistan. Email: mfarooq@upesh.edu.pk
[3] Department of Mathematical Sciences, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, UK. E-mail: as@essex.ac.uk.

of steps [27], where $n$ is the dimension of the problem. Several Lanczos-type algorithms have been designed and among them, the famous conjugate gradient algorithm of Hestenes and Stiefel [25], when the matrix is Hermitian and the bi-conjugate gradient algorithm of Fletcher [22], in the general case. In the last few decades, Lanczos-type algorithms have evolved and different variants have been derived, which can be found in [2, 3, 5, 7, 10, 11, 12, 9, 14, 23, 24, 28, 29, 30, 31, 34, 35, 17].

Lanczos-type algorithms are commonly derived using Formal Orthogonal Polynomials (FOP's), [5]. The connection between the Lanczos algorithm, [27] and orthogonal polynomials, [32] has been studied extensively in [2, 4, 5, 11, 12, 6, 8, 9, 16].

In this paper we will briefly recall recurrence relation $A_{19}$ [17] for the choice of auxiliary polynomial $U_i(x) = x^i$, where $x^i$ is a monic polynomial of degree $i$. Then we will derive expressions for the coefficients of this polynomial for a new choice of $U_i(x) = P_i^{(1)}(x)$ which was not considered before. We will also recall $B_6$ [1] for the same choice of $U_i(x)$. We use the new choice of $A_{19}$ in combination of $B_6$ to derive a new Lanczos-type algorithm $A_{19}/B_6$. This algorithm is then applied to some problems considered in [17, 1, 33], and its performance is compared with that of existing algorithms of the same type namely $A_{12}$, $A_{12}^{new}$ $A_5/B_{10}$ and $A_8/B_{10}$, [2, 17, 33]. The paper is organized as follows. In section 2 we will explain the basic Lanczos process. In section 3 we will discuss the notion of FOPs. Relations $A_{19}$ and $B_6$ are recalled in section 4. A conclusion is given in section 5.

1.1. **The Lanczos Process.** Consider the following system of linear equations,

$$A\mathbf{x} = \mathbf{b}, \tag{1}$$

The basic Lanczos approach for solving SLEs (1), can be explained as follows.

Choose $\mathbf{x}_0$ and $\mathbf{y}$, two arbitrary vectors in $\mathbb{R}^n$, such that $\mathbf{y} \neq 0$, then Lanczos process [27] consists in generating a sequence of vectors $\mathbf{x}_k \in \mathbb{R}^n$, such that

$$(\mathbf{x}_k - \mathbf{x}_0) \in F_k(A, \mathbf{r}_0) = span(\mathbf{r}_0, A\mathbf{r}_0, \ldots, A^{k-1}\mathbf{r}_0), \tag{2}$$

and

$$\mathbf{r}_k = (\mathbf{b} - A\mathbf{x}_k) \perp E_k(A^T, \mathbf{y}) = span(\mathbf{y}, A^T\mathbf{y}, \ldots, (A^T)^{k-1}\mathbf{y}), \tag{3}$$

where $A^T$ is the transpose of matrix $A$.

Equation (2) implies

$$\mathbf{x}_k - \mathbf{x}_0 = -\beta_1 \mathbf{r}_0 - \beta_2 A \mathbf{r}_0 - \cdots - \beta_k A^{k-1} \mathbf{r}_0. \tag{4}$$

Multiplying both sides of (4) by $A$ then adding and subtracting $\mathbf{b}$ on the left hand side of (4) gives

$$\mathbf{r}_k = \mathbf{r}_0 + \beta_1 A \mathbf{r}_0 + \beta_2 A^2 \mathbf{r}_0 + \cdots + \beta_k A^k \mathbf{r}_0. \tag{5}$$

If we set

$$P_k(x) = 1 + \beta_1 x + ... + \beta_k x^k, \tag{6}$$

then we can write from (5)

$$\mathbf{r}_k = P_k(A) \mathbf{r}_0. \tag{7}$$

The polynomials $P_k$ are known as residual polynomials [5]. Another interpretation of the $P_k$ can be found in [15]. From (3), the orthogonality condition implies

$((A^T)^i \mathbf{y}, \mathbf{r}_k) = (\mathbf{y}, A^i \mathbf{r}_k) = (\mathbf{y}, A^i P_k(A) \mathbf{r}_0) = 0$, for $i = 0, ..., k-1$.

Thus, the coefficients $\beta_1$,...,$\beta_k$ form a solution of the following system of linear equations

$$\begin{cases} \beta_1(\mathbf{y}, A\mathbf{r}_0) + \cdots + \beta_k(\mathbf{y}, A^k \mathbf{r}_0) = -(\mathbf{y}, \mathbf{r}_0), \\ \vdots \\ \beta_1((A^T)^{k-1}\mathbf{y}, A\mathbf{r}_0) + \cdots + \beta_k((A^T)^{k-1}\mathbf{y}, A^k \mathbf{r}_0) = -((A^T)^{k-1}\mathbf{y}, \mathbf{r}_0). \end{cases} \tag{8}$$

The scalar products involved in the above system is defined as with the first argument conjugated. If the determinant of the above system is not zero then its solution exists, and thus we can obtain $\mathbf{x}_k$ and $\mathbf{r}_k$ from (4) and (7) respectively. Obviously, in practice, solving the above system directly for increasing values of $k$ is not viable; $k$ is the order of the iterate in the solution process. Now we shall see how to solve the system (8) for increasing value of $k$, that is, if polynomials $P_k$ can be computed recursively.

Now, let $c_i$ be defined as

$$c_i = ((A^T)^i \mathbf{y}, \mathbf{r}_0) = (\mathbf{y}, A^i \mathbf{r}_0), \quad \text{for} \quad i = 1, 2, \ldots,$$

and the linear functional $c$ on the space of polynomials be given by

$$c(x^i) = c_i, \quad \text{for } i = 0, 1 \ldots, \tag{9}$$

so the system (8) can be written as

$$c(x^i P_k(x)) = 0, \quad \text{for } i = 0, 1, \ldots, k-1. \tag{10}$$

These conditions show that $P_k$ is a polynomial of degree at most $k$, corresponding to the linear functional $c$ and normalized by the condition $P_k(0) = 1$. Using normalization condition $P_k(0) = 1$, equation (6) can be written as

$$P_k(x) = 1 + xQ_{k-1}(x),$$

where $Q_{k-1} = \beta_1 + \beta_2 x + ... + \beta_k x^{k-1}$. Replacing $x$ by $A$ and multiplying both sides by $\mathbf{r}_0$ we get

$$P_k(A)\mathbf{r}_0 = \mathbf{r}_0 + AQ_{k-1}(A)\mathbf{r}_0.$$

Now using (7) the above relation becomes

$$\mathbf{r}_k = \mathbf{r}_0 + AQ_{k-1}(A)\mathbf{r}_0,$$

which can also be written as

$$\mathbf{b} - A\mathbf{x}_k = \mathbf{b} - A\mathbf{x}_0 + AQ_{k-1}(A)\mathbf{r}_0.$$

Simplifying and multiplying by $-A^{-1}$ on both sides of the last relation, we get

$$\mathbf{x}_k = \mathbf{x}_0 - Q_{k-1}(A)\mathbf{r}_0,$$

which shows that $\mathbf{x}_k$ can be computed from $\mathbf{r}_k$ recursively without inverting $A$.

1.2. **Formal Orthogonal Polynomials.** The polynomial $P_k(x)$ discussed in the previous section is defined by the following formula [5, 6, 11, 9, 12],

$$P_k(x) = \frac{\begin{vmatrix} 1 & x & \cdots & x^k \\ c_0 & c_1 & \cdots & c_k \\ \vdots & \vdots & & \vdots \\ c_{k-1} & c_k & \cdots & c_{2k-1} \end{vmatrix}}{H_k^{(1)}}, \tag{11}$$

where $H_k^{(1)}$ is called the Hankel determinant [5], which is the determinant of the system (8). This determinant has the following expression:

$$H_k^{(1)} = \begin{vmatrix} c_1 & c_2 & \cdots & c_k \\ c_2 & c_3 & \cdots & c_{k+1} \\ \vdots & \vdots & & \vdots \\ c_k & c_{k+1} & \cdots & c_{2k-1} \end{vmatrix}.$$

Clearly, $P_k$ exists if and only if $H_k^{(1)} \neq 0$. We assume in the following sections that for all $k$, $H_k^{(1)} \neq 0$. If for some $k$, $H_k^{(1)} \approx 0$, then $P_k$ does not exist, and breakdown occurs in the solution process. This breakdown issue is discussed elsewhere, [2, 17, 33].

Let us now define the family of orthogonal polynomials $P_k^{(1)}(x)$ corresponding to the linear functional $c^{(1)}$ where $c^{(1)}$ is define by

$$c^{(1)}(x^i) = c(x^{i+1}) = c_{i+1}, \text{ for } i = 0, 1, \ldots .$$

These polynomials are normalized by the condition that they are monic [5, 6, 11] and are given by the following formula

$$P_k^{(1)}(x) = \frac{\begin{vmatrix} c_1 & \cdots & c_{k+1} \\ \vdots & & \vdots \\ c_k & \cdots & c_{2k} \\ 1 & \cdots & x^k \end{vmatrix}}{H_k^{(1)}}. \tag{12}$$

The necessary and sufficient condition for the existence and uniqueness of $P_k^{(1)}(x)$ is that the Hankel determinant, [5, 6, 11], is different from zero, which is the same condition as for the existence of the polynomial $P_k(x)$.

## 2. RELATIONS $A_{19}$ AND $B_6$

In the following we will recall relations $A_{19}$ [17, 18] and $B_6$ [1, 2] for the choice of auxiliary polynomial $U_i(x) = x^i$, where $x^i$ is a monic polynomial of degree $i$. Then we will derive expressions for the coefficients of these polynomials for the choice of $U_i(x) = P_i^{(1)}(x)$ which was not considered before for relation $A_{19}$. We use the new choice of $A_{19}$ in tandem with $B_6$ to derive a new Lanczos-type algorithm which we call $A_{19}/B_6$.

### 2.1. **Relation** $A_{19}$.

Consider the following recurrence relation investigated in [17]

$$P_k(x) = (A_k x^2 + B_k x + C_k) P_{k-2}^{(1)}(x) + (D_k x + E_k) P_{k-1}(x), \tag{13}$$

where $P_k$, $P_{k-2}^{(1)}$ and $P_{k-1}$ are polynomials of degree $k$, $k-2$ and $k-1$ respectively and $A_k$, $B_k$, $C_k$, $D_k$ and $E_k$ are constants to be determined using the normalization condition $\forall k, P_k(0) = 1$ and orthogonality conditions $(C_1)$ and $(C_2)$ given below

$\forall i = 0, 1 \cdots k-1,\ c(U_i P_k) = 0 \longrightarrow (C_1)$.

$\forall i = 0, 1, \cdots k-1$, $c^{(1)}(U_i P_k^{(1)}) = 0 \longrightarrow (C_2)$.
Using the normalization condition, equation (13) gives

$$1 = E_k + C_k P_{k-2}^{(1)}(0). \tag{14}$$

Multiplying (13) by $U_i$, a polynomial of exact degree $i$ and applying 'c' on both sides we get

$$\begin{aligned} c(U_i P_k) = A_k c(x^2 U_i P_{k-2}^{(1)}) + B_k c(x U_i P_{k-2}^{(1)}) + C_k c(U_i P_{k-2}^{(1)}) \\ + D_k c(x U_i P_{k-1}) + E_k c(U_i P_{k-1}). \end{aligned} \tag{15}$$

Similarly, using $(C_1)$, equation (15) becomes

$$\begin{aligned} A_k c(x^2 U_i P_{k-2}^{(1)}) + B_k c(x U_i P_{k-2}^{(1)}) + C_k c(U_i P_{k-2}^{(1)}) \\ + D_k c(x U_i P_{k-1}) + E_k c(U_i P_{k-1}) = 0. \end{aligned} \tag{16}$$

For $i = 0$, equation (16) becomes $C_k c(U_0 P_{k-2}^{(1)}) = 0$. Since $c(U_0 P_{k-2}^{(1)}) \neq 0$, therefore, $C_k = 0$. Hence from (14), we get $E_k = 1$. The orthogonality condition $(C_1)$ is true for $\forall i = 1, 2, 3, \cdots, k-4$. For $i = k-3$, equation (16) becomes $A_k c(x^2 U_{k-3} P_{k-2}^{(1)}) = 0$. This implies that $A_k = 0$ as $c(x^2 U_{k-3} P_{k-2}^{(1)}) \neq 0$. For $i = k-2$, equation (16) gives

$$B_k c^{(1)}(U_{k-2} P_{k-2}^{(1)}) + D_k c(x U_{k-2} P_{k-1}) = 0. \tag{17}$$

For $i = k-1$, equation (16) gives

$$B_k c^{(1)}(U_{k-1} P_{k-2}^{(1)}) + D_k c(x U_{k-1} P_{k-1}) = -c(U_{k-1} P_{k-1}). \tag{18}$$

If we set $a_{11} = c^{(1)}(U_{k-2} P_{k-2}^{(1)})$, $a_{12} = c(x U_{k-2} P_{k-1})$, $a_{21} = c^{(1)}(U_{k-1} P_{k-2}^{(1)})$, $a_{22} = c(x U_{k-1} P_{k-1})$, $b_1 = 0$, and $b_2 = -c(U_{k-1} P_{k-1})$ then equations (17) and (18) can be written as

$$a_{11} B_k + a_{12} D_k = 0, \tag{19}$$

$$a_{21} B_k + a_{22} D_k = b_2, \tag{20}$$

respectively. If $\Delta_k$ is the determinant of the coefficient matrix of the above system then, $\Delta_k = a_{11} a_{22} - a_{21} a_{12}$. If $\Delta_k \neq 0$ then $B_k = -\frac{b_2 a_{12}}{\Delta_k}$, $D_k = \frac{a_{11} b_2}{\Delta_k}$. Hence,

$$P_k(x) = B_k x P_{k-2}^{(1)}(x) + (D_k x + 1) P_{k-1}(x). \tag{21}$$

Let us apply the recursive formula (21) for computing polynomials $P_k$ in order to find residuals $\mathbf{r}_k$ and the corresponding vector $\mathbf{x}_k$. For this replace $x$ by $A$ and multiply both sides by $\mathbf{r}_0$. Using $\mathbf{r}_k = P_k(A)\mathbf{r}_0$ and $\mathbf{z}_k = P_k^{(1)}(A)\mathbf{r}_0$, we get

$$\mathbf{r}_k = B_k A\mathbf{z}_{k-2} + D_k A\mathbf{r}_{k-1} + \mathbf{r}_{k-1}. \tag{22}$$

Since $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, (22) becomes

$$\mathbf{b} - A\mathbf{x}_k = B_k A\mathbf{z}_{k-2} + D_k A\mathbf{r}_{k-1} + \mathbf{b} - A\mathbf{x}_{k-1}.$$

Multiplying this latter equation by $A^{-1}$ on both sides results in

$$\mathbf{x}_k = \mathbf{x}_{k-1} - B_k\mathbf{z}_{k-2} - D_k\mathbf{r}_{k-1}, \tag{23}$$

where coefficients $B_k$ and $D_k$ can be identified as $B_k = -\frac{b_2 a_{12}}{\Delta_k}$, $D_k = \frac{a_{11} b_2}{\Delta_k}$, respectively. So, now we choose the polynomial $U_i(x)$.

2.1.1. *Case-I: When $U_i(x) = x^i$.* In the previous section we discussed $A_{19}$ for general auxiliary polynomial $U_i$. Here we recall from [17] briefly the same relation $A_{19}$ by taking $U_i = x^i$. In [17] we have $\Delta_k = a_{11}a_{22} - a_{21}a_{12}$, where $a_{11} = c^{(1)}(x^{k-2}P_{k-2}^{(1)})$, $a_{12} = c(x^{k-1}P_{k-1})$, $a_{21} = c^{(1)}(x^{k-1}P_{k-2}^{(1)})$, $a_{22} = c(x^k P_{k-1})$, $b_1 = 0$, and $b_2 = -c(x^{k-1}P_{k-1})$. If $\Delta_k \neq 0$ then coefficients $B_k$ and $D_k$ appearing in (21) are as above. Since we know that

$$\begin{cases} c(x^k P_k) = ((A^T)^k\mathbf{y}, P_k(A)\mathbf{r}_0) = (\mathbf{y}_k, \mathbf{r}_k) \ \ \textit{and} \\ c(x^k P_k^{(1)}) = ((A^T)^k\mathbf{y}, P_k^{(1)}(A)\mathbf{r}_0) = (\mathbf{y}_k, \mathbf{z}_k), \\ \text{ with } \quad \mathbf{y}_k = A^T\mathbf{y}_{k-1}, \end{cases} \tag{24}$$

using (24), we can write $a_{11} = c^{(1)}(x^{k-2}P_{k-2}^{(1)}) = (\mathbf{y}_{k-1}, \mathbf{z}_{k-2})$,
$a_{12} = c(x^{k-1}P_{k-1}) = (\mathbf{y}_{k-1}, \mathbf{r}_{k-1})$, $a_{21} = c^{(1)}(x^{k-1}P_{k-2}^{(1)}) = (\mathbf{y}_k, \mathbf{z}_{k-2})$,
$a_{22} = c(x^k P_{k-1}) = (\mathbf{y}_k, \mathbf{r}_{k-1})$, $b_1 = 0$, and $b_2 = -c(x^{k-1}P_{k-1}) =$
$-(\mathbf{y}_{k-1}, \mathbf{r}_{k-1})$.

Now, since all of the above relations are only valid for $k \geq 3$, to evaluate (22) and (23) recursively, we need to evaluate $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{z}_1$ and $\mathbf{z}_2$, which are necessary, differently. These values are determined in detail in [1, 17]. They are recalled briefly here, however, for completeness, as follows.
$\mathbf{r}_1 = \mathbf{r}_0 - (\frac{c_0}{c_1})A\mathbf{r}_0$, $\mathbf{x}_1 = \mathbf{x}_0 + (\frac{c_0}{c_1})\mathbf{r}_0$ where $c_i = (\mathbf{y}, A^i\mathbf{r}_0)$,
$\mathbf{r}_2 = \mathbf{r}_0 - \alpha A\mathbf{r}_0 + \beta A^2\mathbf{r}_0$, $\mathbf{x}_2 = \mathbf{x}_0 + \alpha\mathbf{r}_0 - \beta A\mathbf{r}_0$,
$\mathbf{z}_1 = A\mathbf{r}_0 - (\frac{c_2}{c_1})\mathbf{r}_0$, $\mathbf{z}_2 = A^2\mathbf{r}_0 - \alpha_1 A\mathbf{r}_0 + \beta_1\mathbf{r}_0$.
Also $\tilde{\mathbf{z}_1} = A\tilde{\mathbf{z}_0} - (\frac{c_2}{c_1})\tilde{\mathbf{z}_0}$, $\tilde{\mathbf{z}_2} = A^2\tilde{\mathbf{z}_0} - \alpha_1 A\tilde{\mathbf{z}_0} + \beta_1\tilde{\mathbf{z}_0}$,

where $\delta = c_1c_3 - c_2^2$, $\alpha = \frac{c_0c_3-c_1c_2}{\delta}$, $\beta = \frac{c_0c_2-c_1^2}{\delta}$,$\delta_1 = c_1c_3 - c_2^2$, $\alpha_1 = \frac{c_1c_4-c_2c_3}{\delta_1}$, and $\beta_1 = \frac{c_2c_4-c_3^2}{\delta_1}$.

2.1.2. *Case-II: When* $U_i(x) = P_i^{(1)}(x)$. In this section, we derive $A_{19}$ for a different choice of $U_i(x)$ which was not considered before. All the coefficients involved in $A_{19}$ have completely different expressions for this new choice of $U_i(x)$ as explained below. For $U_i(x) = P_i^{(1)}(x)$ all of the above expressions will have the following form:
$a_{11} = c^{(1)}(P_{k-2}^{(1)}P_{k-2}^{(1)})$, $a_{12} = c(xP_{k-2}^{(1)}P_{k-1})$, $a_{21} = c^{(1)}(P_{k-1}^{(1)}P_{k-2}^{(1)})$, $a_{22} = c(xP_{k-1}^{(1)}P_{k-1})$, $b_1 = 0$, and $b_2 = -c(P_{k-1}^{(1)}P_{k-1})$. Using

$$\begin{cases} c^{(1)}(P_{k-1}^{(1)}P_{k-2}^{(1)}) = 0, \\ \mathbf{z}_k = P_k^{(1)}(A)\mathbf{r}_0,\ \tilde{\mathbf{z}}_k = P_k^{(1)}(A^T)\tilde{\mathbf{z}}_0,\ \mathbf{r}_k = P_k(A)\mathbf{r}_0,\ \textit{and} \\ c(U_kP_k) = (y, U_k(A)P_k(A)r_0) = (U_k(A^T)y, P_k(A)r_0) = (\tilde{z}_k, r_k) \\ [\textit{note}\ \tilde{\mathbf{z}}_0 = y], \end{cases} \tag{25}$$

we get $a_{21} = 0$, $a_{11} = c^{(1)}(P_{k-2}^{(1)}P_{k-2}^{(1)}) = (\tilde{\mathbf{z}}_{k-2}, A\mathbf{z}_{k-2})$, $a_{12} = c(xP_{k-2}^{(1)}P_{k-1})$ $= (\tilde{\mathbf{z}}_{k-2}, A\mathbf{r}_{k-1})$, $a_{22} = c(xP_{k-1}^{(1)}P_{k-1}) = (\tilde{\mathbf{z}}_{k-1}, A\mathbf{r}_{k-1})$, $b_1 = 0$, $b_2 = -c(P_{k-1}^{(1)}P_{k-1}) = -(\tilde{\mathbf{z}}_{k-1}, \mathbf{r}_{k-1})$, $\Delta_k = a_{11}a_{22}$, $B_k = -\frac{b_2a_{12}}{\Delta_k}$, and $D_k = \frac{b_2}{a_{22}}$. Hence, after evaluating the coefficients $A_k$, $B_k$, $C_k$, $\Delta_k$ and $E_k$ for the choice $U_i(x) = P_i^{(1)}(x)$ equation(13) reduces to

$$P_k(x) = B_kxP_{k-2}^{(1)}(x) + (D_kx + 1)P_{k-1}(x). \tag{26}$$

Replacing $x$ by $A$ and multiplying both sides by $\mathbf{r}_0$ and using $\mathbf{r}_k = P_k(A)\mathbf{r}_0$, and $\mathbf{z}_k = P_k^{(1)}(A)\mathbf{r}_0$ we get

$$\mathbf{r}_k = B_kA\mathbf{z}_{k-2} + D_kA\mathbf{r}_{k-1} + \mathbf{r}_{k-1}. \tag{27}$$

Since $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, (14) becomes

$$\mathbf{x}_k = \mathbf{x}_{k-1} - B_k\mathbf{z}_{k-2} - D_k\mathbf{r}_{k-1}. \tag{28}$$

2.2. **Relation** $B_6$**.** Consider the following recurrence relation investigated in [1]

$$P_k^{(1)}(x) = (A_kx^2 + B_kx + C_k)P_{k-2}^{(1)}(x) + (D_kx + E_k)P_{k-1}^{(1)}(x), \tag{29}$$

where $P_k^{(1)}$, $P_{k-2}^{(1)}$ and $P_{k-1}^{(1)}$ are polynomials of degree $k$, $k-2$ and $k-1$ respectively and $A_k$, $B_k$, $C_k$, $D_k$ and $E_k$ are constants to be determined as already discussed in[1, 2] for the choices $x^i$, $P_k(x)$, $P_k^{(1)}(x)$. We discuss and recall $B_6$ for general auxiliary polynomial $U_i$, a polynomial of exact degree $i$. Multiply (29) by $U_i$ and apply $c^{(1)}$ on both sides to get

$$\begin{aligned} c^{(1)}(U_iP_k^{(1)}) &= A_kc^{(1)}(x^2U_iP_{k-2}^{(1)}) + B_kc^{(1)}(xU_iP_{k-2}^{(1)}) \\ &+C_kc^{(1)}(U_iP_{k-2}^{(1)}) + D_kc^{(1)}(xU_iP_{k-1}^{(1)}) + E_kc^{(1)}(U_iP_{k-1}^{(1)}). \end{aligned} \tag{30}$$

Using $(C_2)$ we get

$$\begin{aligned} A_kc^{(1)}(x^2U_iP_{k-2}^{(1)}) + B_kc^{(1)}(xU_iP_{k-2}^{(1)}) + C_kc^{(1)}(U_iP_{k-2}^{(1)}) \\ +D_kc^{(1)}(xU_iP_{k-1}^{(1)}) + E_kc^{(1)}(U_iP_{k-1}^{(1)}) = 0. \end{aligned} \tag{31}$$

The orthogonality condition $(C_2)$ is true for $\forall i = 0, 1, 2, \cdots k-5$.
For $i = k-4$, we get $A_kc^{(1)}(x^2U_{k-4}P_{k-2}^{(1)}) = 0$, which implies that $A_k = 0$ as $c^{(1)}(x^2U_{k-4}P_{k-2}^{(1)}) \neq 0$. But $P_k^{(1)}$ is a monic polynomial of degree $k$; so $D_k = 1$. For $i = k-3$, we get $B_kc^{(1)}(xU_{k-3}P_{k-2}^{(1)}) = 0$. Since $c^{(1)}(xU_{k-3}P_{k-2}^{(1)}) \neq 0$, $B_k = 0$.
For $i = k-2$, we have $c^{(1)}(xU_{k-2}P_{k-1}^{(1)}) + C_kc^{(1)}(U_{k-2}P_{k-2}^{(1)}) = 0$ which implies that

$$C_k = -\frac{c^{(1)}(xU_{k-2}P_{k-1}^{(1)})}{c^{(1)}(U_{k-2}P_{k-2}^{(1)})}. \tag{32}$$

For $i = k-1$, we get

$$c^{(1)}(xU_{k-1}P_{k-1}^{(1)}) + C_kc^{(1)}(U_{k-1}P_{k-2}^{(1)}) + E_kc^{(1)}(U_{k-1}P_{k-1}^{(1)}) = 0.$$

Since $c^{(1)}(U_{k-1}P_{k-1}^{(1)}) \neq 0$,

$$E_k = \frac{-c^{(1)}(xU_{k-1}P_{k-1}^{(1)}) - C_kc^{(1)}(U_{k-1}P_{k-2}^{(1)})}{c^{(1)}(U_{k-1}P_{k-1}^{(1)})}. \tag{33}$$

Hence (29) becomes

$$P_k^{(1)}(x) = C_kP_{k-2}^{(1)}(x) + (x + E_k)P_{k-1}^{(1)}(x),$$

where

$$C_k = -\frac{c^{(1)}(xU_{k-2}P_{k-1}^{(1)})}{c^{(1)}(U_{k-2}P_{k-2}^{(1)})},$$

and

$$E_k = \frac{-c^{(1)}(xU_{k-1}P_{k-1}^{(1)}) - C_kc^{(1)}(U_{k-1}P_{k-2}^{(1)})}{c^{(1)}(U_{k-1}P_{k-1}^{(1)})}.$$

2.2.1. *Relation* $B_6$ *when* $U_i(x) = P_i^{(1)}(x)$. In this case, (32) and (33) become $C_k = -\frac{c(x^2 P_{k-2}^{(1)} P_{k-1}^{(1)})}{c(x P_{k-2}^{(1)} P_{k-2}^{(1)})}$, and
$E_k = \frac{-c^{(1)}(x P_{k-1}^{(1)} P_{k-1}^{(1)}) - C_k c^{(1)}(P_{k-1}^{(1)} P_{k-2}^{(1)})}{c^{(1)}(P_{k-1}^{(1)} P_{k-1}^{(1)})}, respectively.$ Using

$$\begin{cases} c^{(1)}(P_{k-1}^{(1)} P_{k-2}^{(1)}) = 0, \\ \mathbf{z}_k = P_k^{(1)}(A)\mathbf{r}_0,\ \tilde{\mathbf{z}}_k = P_k^{(1)}(A^T)\tilde{\mathbf{z}}_0,\ \mathbf{r}_k = P_k(A)\mathbf{r}_0,\ and \\ \ c(U_k P_k) = (y, U_k(A)P_k(A)r_0) = (U_k(A^T)y, P_k(A)r_0) = (\tilde{z}_k, r_k), \\ \ \ [note\ \tilde{\mathbf{z}}_0 = y], \end{cases} \tag{34}$$

we get $C_k = -\frac{c(x^2 P_{k-2}^{(1)} P_{k-1}^{(1)})}{c(x P_{k-2}^{(1)} P_{k-2}^{(1)})} = -\frac{(\tilde{\mathbf{z}}_{k-2}, A^2\mathbf{z}_{k-1})}{(\tilde{\mathbf{z}}_{k-2}, A\mathbf{z}_{k-2})}$, and
$E_k = -\frac{c(x^2 P_{k-1}^{(1)} P_{k-1}^{(1)})}{c(x P_{k-1}^{(1)} P_{k-1}^{(1)})} = -\frac{(\tilde{\mathbf{z}}_{k-1}, A^2\mathbf{z}_{k-1})}{(\tilde{\mathbf{z}}_{k-1}, A\mathbf{z}_{k-1})}$.
Hence after evaluating the coefficients $A_k$, $B_k$, $C_k$, $D_k$ and $E_k$ for the choice $U_i(x) = P_i^{(1)}(x)$ equation (29) reduces to

$$P_k^{(1)}(x) = C_k P_{k-2}^{(1)}(x) + (x + E_k) P_{k-1}^{(1)}(x). \tag{35}$$

Replacing $x$ by $A$, multiplying by $\mathbf{r}_0$ and using $\mathbf{z}_k = P_k^{(1)}(A)\mathbf{r}_0$ we get

$$\mathbf{z}_k = C_k\mathbf{z}_{k-2} + A\mathbf{z}_{k-1} + E_k\mathbf{z}_{k-1}.$$

Replacing $x$ by $A^T$ and multiply by $\tilde{\mathbf{z}}_0 = \mathbf{y}$ and using $\tilde{\mathbf{z}}_k = P_k^{(1)}(A^T)\tilde{\mathbf{z}}_0$, we get

$$\tilde{\mathbf{z}}_k = C_k\tilde{\mathbf{z}}_{k-2} + A^T\tilde{\mathbf{z}}_{k-1} + E_k\tilde{\mathbf{z}}_{k-1}.$$

2.3. **Algorithm** $A_{19}/B_6$**.** We now consider the combination of $A_{19}$ and $B_6$ for the choice $U_i(x) = P_i^{(1)}(x)$ which, as said earlier, was never considered before. The new algorithm is called $A_{19}/B_6$ and its pseudo-code is given below as Algorithm 1.

**Algorithm 1** Lanczos-type Algorithm $A_{19}/B_6$.

```
Choose $\mathbf{x}_0$ and $\mathbf{y}$ such that $\mathbf{y} \neq 0$.
Set $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$, $\tilde{\mathbf{z}}_0 = \mathbf{y}$,
$\mathbf{z}_0 = \mathbf{r}_0$.
$\mathbf{p} = A\mathbf{r}_0$, $\mathbf{p}_1 = A\mathbf{p}$, $\mathbf{p}_2 = A\mathbf{p}_1$, $\mathbf{p}_3 = A\mathbf{p}_2$,
$c_0 = (\mathbf{y}, \mathbf{r}_0)$, $c_1 = (\mathbf{y}, \mathbf{p})$, $c_2 = (\mathbf{y}, \mathbf{p}_1)$,
$c_3 = (\mathbf{y}, \mathbf{p}_2)$, $c_4 = (\mathbf{y}, \mathbf{p}_3)$, $\delta = c_1c_3 - c_2^2$,
$\alpha = \frac{c_0c_3 - c_1c_2}{\delta}$, $\beta = \frac{c_0c_2 - c_1^2}{\delta}$, $\delta_1 = c_1c_3 - c_2^2$,
$\alpha_1 = \frac{c_1c_4 - c_2c_3}{\delta_1}$, $\beta_1 = \frac{c_2c_4 - c_3^2}{\delta_1}$,
$\mathbf{r}_1 = \mathbf{r}_0 - (\frac{c_0}{c_1})\mathbf{p}$, $\mathbf{x}_1 = \mathbf{x}_0 + (\frac{c_0}{c_1})\mathbf{r}_0$,
$\mathbf{r}_2 = \mathbf{r}_0 - \alpha\mathbf{p} + \beta\mathbf{p}_1$,
$\mathbf{x}_2 = \mathbf{x}_0 + \alpha\mathbf{r}_0 - \beta\mathbf{p}$,
$\mathbf{z}_1 = \mathbf{p} - (\frac{c_2}{c_1})\mathbf{r}_0$, $\mathbf{z}_2 = \mathbf{p}_1 - \alpha_1\mathbf{p} + \beta_1\mathbf{r}_0$,
$\mathbf{y}_1 = A^T\mathbf{y}$, $\mathbf{y}_2 = A^T\mathbf{y}_1$,
$\tilde{\mathbf{z}}_1 = \mathbf{y}_1 - (\frac{c_2}{c_1})\tilde{\mathbf{z}}_0$,
$\tilde{\mathbf{z}}_2 = \mathbf{y}_2 - \alpha_1\mathbf{y}_1 + \beta_1\tilde{\mathbf{z}}_0$,
for k=3,4,... do
    $q_1 = Ar_{k-1}$, $q_2 = Az_{k-1}$, $q_3 = Aq_2$, $q_4 = Az_{k-2}$, $s = A^T\tilde{\mathbf{z}}_{k-1}$
    $a_{11} = c^{(1)}(P_{k-2}^{(1)}P_{k-2}^{(1)}) = (\tilde{\mathbf{z}}_{k-2}, q_4)$
    $a_{12} = c(xP_{k-2}^{(1)}P_{k-1}) = (\tilde{\mathbf{z}}_{k-2}, q_1)$
    $a_{22} = c(xP_{k-1}^{(1)}P_{k-1}) = (\tilde{\mathbf{z}}_{k-1}, q_1)$
    $b_2 = -c(P_{k-1}^{(1)}P_{k-1}) = -(\tilde{\mathbf{z}}_{k-1}, \mathbf{r}_{k-1})$
    $\Delta_k = a_{11}a_{22}$,
    if $\Delta_k \leq \epsilon$
        print "ghost-type breakdown"
        stop.
    end if
    $B_k = -\frac{b_2a_{12}}{\Delta_k}$, $D_k = \frac{b_2}{a_{22}}$
    $\mathbf{r}_k = B_kq_4 + D_kq_1 + \mathbf{r}_{k-1}$
    $\mathbf{x}_k = \mathbf{x}_{k-1} - B_k\mathbf{z}_{k-2} - D_k\mathbf{r}_{k-1}$
    if $||\mathbf{r}_k|| > \epsilon$ then
        $C_k = -\frac{c(x^2P_{k-2}^{(1)}P_{k-1}^{(1)})}{c(xP_{k-2}^{(1)}P_{k-2}^{(1)})} = -\frac{(\tilde{\mathbf{z}}_{k-2}, q_3)}{(\tilde{\mathbf{z}}_{k-2}, q_4)}$
        $E_k = -\frac{c(x^2P_{k-1}^{(1)}P_{k-1}^{(1)})}{c(xP_{k-1}^{(1)}P_{k-1}^{(1)})} = -\frac{(\tilde{\mathbf{z}}_{k-1}, q_3)}{(\tilde{\mathbf{z}}_{k-1}, q_2)}$
        $\mathbf{z}_k = C_k\mathbf{z}_{k-2} + q_2 + E_k\mathbf{z}_{k-1}$
        $\tilde{\mathbf{z}}_k = C_k\tilde{\mathbf{z}}_{k-2} + s + E_k\tilde{\mathbf{z}}_{k-1}$
    else
        $\mathbf{x} = \mathbf{x}_k$
        stop
    end if
end for
```

Algorithm 1 has been implemented in Matlab and tested on the following problem which was considered in [17, 1, 33, 21]. This problem arises in the 5-point discretisation of the operator $\frac{-d^2}{dx^2} - \frac{d^2}{dy^2} + \gamma\frac{d}{dx}$ on a rectangular region, [17, 1]. Comparative results on instances of the problem $A\mathbf{x} = \mathbf{b}$ ranging from dimension 10 to 900 for parameter $\delta$ taking value 0.0 and 0.2 and for the tolerance $\epsilon = 10^{-05}$ are recorded in Tables 1 and 2.

$$A = \begin{pmatrix} B & -I & \cdots & \cdots & 0 \\ -I & B & -I & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & -I & B & -I \\ 0 & \cdots & \cdots & -I & B \end{pmatrix},$$

with

$$B = \begin{pmatrix} 4 & \alpha & \cdots & \cdots & 0 \\ \beta & 4 & \alpha & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \beta & 4 & \alpha \\ 0 & \cdots & & \beta & 4 \end{pmatrix},$$

and $\alpha = -1 + \delta$, $\beta = -1 - \delta$. When $\delta = 0$, the matrix of coefficients A is symmetric and the problem is easy to solve. For $\delta = 0.2$ the matrix A is non-symmetric and the problem is comparatively harder as the region is not a regular mesh. The dimension of the matrix $B = 10$. The right hand side $\mathbf{b}$ is taken to be $\mathbf{b} = A\mathbf{x}$, where $\mathbf{x} = (1, 1, \ldots, 1)^T$, is the solution of the system.

TABLE 1. $A_{19}/B_6$ versus $A_{12}$, $A_{12}^{new}$, $A_5/B_{10}$ and $A_8/B_{10}$ for problems of different dimensions when $\delta = 0$

| | $A_5/B_{10}$ | | $A_8/B_{10}$ | | $A_{12}$ | | $A_{12}^{new}$ | | $A_{19}/B_6$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) |
| 10 | $2.2940E^{-13}$ | 0.010854 | $1.7704E^{-13}$ | 0.010376 | $4.9623E^{-13}$ | 0.048924 | $2.9118E^{-13}$ | 0.020067 | $6.8468E^{-13}$ | 0.008025 |
| 20 | $2.5256E^{-14}$ | 0.011083 | $1.7489E^{-13}$ | 0.010333 | $1.7536E^{-13}$ | 0.048976 | $2.4453E^{-15}$ | 0.020012 | $6.1935E^{-07}$ | 0.008509 |
| 30 | $3.9026E^{-09}$ | 0.011525 | $4.9472E^{-09}$ | 0.010885 | $5.4705E^{-08}$ | 0.049626 | $2.5346E^{-10}$ | 0.021378 | $8.5523E^{-06}$ | 0.009085 |
| 40 | $1.4770E^{-10}$ | 0.011533 | $8.4658E^{-10}$ | 0.011027 | $1.4776E^{-08}$ | 0.049785 | $3.6924E^{-11}$ | 0.021413 | $8.1290E^{-06}$ | 0.010240 |
| 50 | $1.9959E^{-06}$ | 0.012044 | $1.3598E^{-06}$ | 0.011429 | $4.7994E^{-06}$ | 0.051143 | $1.2732E^{-06}$ | 0.022533 | $9.2021E^{-06}$ | 0.010890 |
| 60 | $9.1910E^{-06}$ | 0.012473 | $3.7470E^{-06}$ | 0.011487 | $5.0010E^{-06}$ | 0.051385 | $2.3592E^{-06}$ | 0.022561 | $3.1422E^{-06}$ | 0.010661 |
| 70 | $4.9035E^{-06}$ | 0.013022 | $4.2579E^{-06}$ | 0.012160 | $1.3781E^{-06}$ | 0.052743 | $5.1279E^{-07}$ | 0.023865 | $4.5622E^{-06}$ | 0.011104 |
| 80 | $4.4311E^{-06}$ | 0.013973 | $7.7199E^{-06}$ | 0.013356 | $7.5581E^{-06}$ | 0.052522 | $3.5448E^{-06}$ | 0.023640 | $7.2687E^{-06}$ | 0.011604 |
| 90 | $NaN$ | | $9.2478E^{-06}$ | 0.019182 | $6.2686E^{-06}$ | 0.055501 | $4.4189E^{-06}$ | 0.024360 | $3.4159E^{-06}$ | 0.012699 |
| 100 | $1.1889E^{-06}$ | 0.013331 | $3.1695E^{-06}$ | 0.012546 | $8.9530E^{-07}$ | 0.052106 | $2.3809E^{-07}$ | 0.023172 | $5.8577E^{-06}$ | 0.012053 |
| 200 | $NaN$ | | $NaN$ | | $NaN$ | | $8.2198E^{-06}$ | 0.041164 | $5.8982E^{-06}$ | 0.026923 |
| 300 | $NaN$ | | $NaN$ | | $NaN$ | | $7.3650E^{-06}$ | 0.083002 | $3.6483E^{-06}$ | 0.055710 |
| 400 | $NaN$ | | $NaN$ | | $NaN$ | | $8.2378E^{-06}$ | 0.121768 | $8.9684E^{-06}$ | 0.130579 |
| 500 | $NaN$ | | $NaN$ | | $NaN$ | | $9.8283E^{-06}$ | 0.990127 | $7.5932E^{-06}$ | 0.260776 |
| 600 | $NaN$ | | $NaN$ | | $NaN$ | | $9.8207E^{-06}$ | 1.574158 | $9.8773E^{-06}$ | 0.466735 |
| 700 | $NaN$ | | $NaN$ | | $NaN$ | | $9.4625E^{-06}$ | 2.854015 | $9.7121E^{-06}$ | 0.720233 |
| 800 | $NaN$ | | $NaN$ | | $NaN$ | | $9.1387E^{-06}$ | 4.476857 | $8.6149E^{-06}$ | 1.254427 |
| 900 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $7.4319E^{-06}$ | 4.066442 |

The results show that for $\delta = 0$ algorithm $A_{19}/B_6$ solved the given problems for dimensions up to 900 while the existing three algorithms namely $A_5/B_{10}$, $A_8/B_{10}$ and $A_{12}$ failed on systems of dimension $n > 100$.

TABLE 2. $A_{19}/B_6$ versus $A_{12}$, $A_{12}^{new}$, $A_5/B_{10}$ and $A_8/B_{10}$ for problems of different dimensions when $\delta = 0.2$

| | $A_5/B_{10}$ | | $A_8/B_{10}$ | | $A_{12}$ | | $A_{12}^{new}$ | | $A_{19}/B_6$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) |
| 10 | $9.8216E^{-10}$ | 0.017858 | $2.3567E^{-10}$ | 0.010808 | $2.0583E^{-08}$ | 0.049232 | $4.7266E^{-08}$ | 0.021279 | $7.5451E^{-06}$ | 0.017863 |
| 20 | $4.1778E^{-11}$ | 0.011341 | $5.8526E^{-11}$ | 0.010930 | $6.3915E^{-10}$ | 0.049101 | $5.9892E^{-10}$ | 0.021293 | $3.7108E^{-06}$ | 0.020886 |
| 30 | $2.6438E^{-06}$ | 0.012366 | $5.9072E^{-06}$ | 0.012117 | $5.9403E^{-06}$ | 0.050672 | $6.8627E^{-06}$ | 0.022881 | $9.2819E^{-06}$ | 0.021169 |
| 40 | $NaN$ | | $NaN$ | | $7.6080E^{-06}$ | 0.051437 | $7.8688E^{-06}$ | 0.023776 | $6.8914E^{-06}$ | 0.023724 |
| 50 | $NaN$ | | $NaN$ | | $8.8143E^{-06}$ | 0.066431 | $5.0100E^{-06}$ | 0.028116 | $7.2611E^{-06}$ | 0.022813 |
| 60 | $NaN$ | | $NaN$ | | $NaN$ | | $2.6424E^{-06}$ | 0.024839 | $5.9941E^{-06}$ | 0.025928 |
| 70 | $NaN$ | | $NaN$ | | $NaN$ | | $9.9853E^{-06}$ | 0.237743 | $9.3422E^{-06}$ | 0.024190 |
| 80 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $3.8215E^{-06}$ | 0.025606 |
| 90 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $7.7488E^{-06}$ | 0.037380 |
| 100 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $7.7186E^{-06}$ | 0.045345 |
| 200 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $3.5339E^{-06}$ | 0.052006 |
| 300 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $6.5440E^{-06}$ | 0.136388 |
| 400 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $7.2847E^{-06}$ | 0.259785 |
| 500 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $2.8823E^{-06}$ | 0.278282 |
| 600 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $8.9049E^{-06}$ | 0.445157 |

As can be seen in Table 2, for $\delta = 0.2$, algorithm $A_{19}/B_6$ solved the given problems for dimensions up to 500 while algorithms $A_5/B_{10}$, and $A_8/B_{10}$ failed for $n = 40$ and $A_{12}$, $A_{12}^{new}$ failed for $n = 60$ and above.
If we decrease the tolerance $\epsilon$ from $10^{-05}$ to $10^{-013}$ the numerical results are strongly in favor of $A_{19}/B_6$ which is clear from Table 3 and Table 4.

Table 3. $A_{19}/B_6$ versus $A_{12}$, $A_{12}^{new}$, $A_5/B_{10}$ and $A_8/B_{10}$ for problems of different dimensions when $\delta = 0$

| | $A_5/B_{10}$ | | $A_8/B_{10}$ | | $A_{12}$ | | $A_{12}^{new}$ | | $A_{19}/B_6$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) |
| 10 | $4.5196e^{-015}$ | 0.010969 | $4.4052e^{-014}$ | 0.010418 | $3.4389e^{-014}$ | 0.048920 | $2.9118e^{-015}$ | 0.020013 | $9.4527e^{-014}$ | 0.019143 |
| 20 | $2.5256e^{-014}$ | 0.010665 | $1.6217e^{-014}$ | 0.010309 | $4.4538e^{-014}$ | 0.048298 | $2.4453e^{-015}$ | 0.019989 | $8.2368e^{-014}$ | 0.019618 |
| 30 | $NaN$ | | $NaN$ | | $NaN$ | | $9.2583e^{-014}$ | 0.030612 | $3.2875e^{-014}$ | 0.023630 |
| 40 | $NaN$ | | $NaN$ | | $NaN$ | | $8.4447e^{-014}$ | 0.027055 | $2.2496e^{-014}$ | 0.027095 |
| 50 | $NaN$ | | $NaN$ | | $NaN$ | | $4.4856e^{-014}$ | 0.057526 | $1.5384e^{-014}$ | 0.027677 |
| 60 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $3.9895e^{-014}$ | 0.025539 |
| 70 | $NaN$ | | $NaN$ | | $NaN$ | | $9.0212e^{-014}$ | 0.047472 | $2.0157e^{-014}$ | 0.027536 |
| 80 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $7.3023e^{-014}$ | 0.028718 |
| 90 | $NaN$ | | $NaN$ | | $NaN$ | | $4.5477e^{-014}$ | 0.072918 | $6.4488e^{-014}$ | 0.032048 |
| 100 | $NaN$ | | $NaN$ | | $NaN$ | | $6.8764e^{-014}$ | 0.193226 | $4.8541e^{-014}$ | 0.030108 |
| 200 | $NaN$ | | $NaN$ | | $NaN$ | | $7.6191e^{-014}$ | 0.359243 | $4.8439e^{-014}$ | 0.076182 |
| 300 | $NaN$ | | $NaN$ | | $NaN$ | | $4.3659e^{-014}$ | 1.027307 | $6.8328e^{-014}$ | 0.227479 |
| 400 | $NaN$ | | $NaN$ | | $NaN$ | | $9.7388e^{-014}$ | 3.493297 | $7.2821e^{-014}$ | 0.517246 |
| 500 | $NaN$ | | $NaN$ | | $NaN$ | | $9.5239e^{-014}$ | 10.783390 | $6.3551e^{-014}$ | 1.951565 |

The results show that for $\epsilon = 10^{-013}$ and $\delta = 0$ algorithms $A_{19}/B_6$ and $A_{12}^{new}$ solved the given problem for dimensions up to 500 while $A_5/B_{10}$, $A_8/B_{10}$ and $A_{12}$ failed for $n = 30$ and above.

Table 4. $A_{19}/B_6$ versus $A_{12}$, $A_{12}^{new}$, $A_5/B_{10}$ and $A_8/B_{10}$ for problems of different dimensions when $\delta = 0.2$

| | $A_5/B_{10}$ | | $A_8/B_{10}$ | | $A_{12}$ | | $A_{12}^{new}$ | | $A_{19}/B_6$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) |
| 10 | $1.4521E^{-14}$ | 0.012071 | $4.7905E^{-14}$ | 0.011465 | $2.9806E^{-14}$ | 0.050647 | $NaN$ | | $2.4294E^{-14}$ | 0.020081 |
| 20 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $6.0869E^{-14}$ | 0.028402 |
| 30 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $5.1766E^{-14}$ | 0.028676 |
| 40 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $5.1502E^{-14}$ | 0.032685 |
| 50 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $8.9242E^{-14}$ | 0.031879 |
| 60 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $1.9212E^{-14}$ | 0.037306 |
| 70 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $5.5211E^{-14}$ | 0.048051 |
| 80 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $9.8420E^{-14}$ | 0.049704 |
| 90 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $5.0930E^{-14}$ | 0.061245 |
| 100 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $9.0537E^{-14}$ | 0.069353 |
| 200 | $NaN$ | | $NaN$ | | $NaN$ | | $NaN$ | | $1.0460E^{-14}$ | 0.126791 |

Again, for $\epsilon = 10^{-013}$ and $\delta = 0.2$ algorithm $A_{19}/B_6$ solved the given problems up to dimension 200 while the other algorithms failed for $n = 10$ and above. The obvious reason is breakdown, [19, 20]. Since all these algorithms consist of recursively computing $P_k$ and $P_k^{(1)}$, which involves the calculation of some scalar products appearing as denominators and numerators of the coefficient of the recurrence relationships, when any of the denominators become very small, as small as, for instance $2.3879 \times 10^{-014}$, breakdown occurs and the algorithms fail. This breakdown issue is being investigated further and any finding will be reported in forthcoming papers.

According to [1, 17, 21, 33], algorithms $A_{12}$, $A_{12}^{new}$, $A_5/B_{10}$ and $A_8/B_{10}$ are considered as the most robust Lanczos-type algorithms. We have now compared our new algorithm with these algorithms on a standard problem considered in this paper and elsewhere. Our results show that algorithm $A_{19}/B_6$ is faster through out, and more robust overall.

## 3. Conclusion

In this paper, we derived the recurrence relation $A_{19}$ [17] and recalled $B_6$ [1] both using the general auxilliary polynomial $U_i(x)$. We used $A_{19}$ in tandem with $B_6$ to derive a new Lanczos-type algorithm $A_{19}/B_6$. This new algorithm has been applied to a number of instances of some standard test problem considered in [17, 1, 33] and elsewhere. The performance of this algorithm is compared to that of existing and well established algorithms of the same type namely, $A_{12}$, $A_{12}^{new}$, $A_5/B_{10}$ and $A_8/B_{10}$, [2, 17, 33]. Numerical results are strongly in favour of the new algorithm $A_{19}/B_6$.

## References

[1] C. Baheux. *Algorithmes d'implementation de la méthode de Lanczos.* PhD thesis, University of Lille 1, France, 1994.

[2] C. Baheux. New Implementations of Lanczos Method. *Journal of Computational and Applied Mathematics*, 57:3–15, 1995.

[3] A. Bjôrck, T. Elfving, and Z. Strakos. Stability of Conjugate Gradient and Lanczos Methods for Linear Least Squares Problems. *SIAM Journal of Matrix Analysis and Application*, 19:720–736, 1998.

[4] C. Brezinski. *Padé-Type Approximation and General Orthogonal Polynomials, Internat. Ser. Nuner. Math. 50.* Birkhäuser, Basel, 1980.

[5] C. Brezinski and H. Sadok. Lanczos-type algorithms for solving systems of linear equations. *Applied Numerical Mathematics*, 11:443–473, 1993.

[6] C. Brezinski and M. R. Zaglia. A new presentation of orthogonal polynomials with applications to their computation. *Numerical Algorithms*, 1:207–222, 1991.

[7] C. Brezinski and M. R. Zaglia. Hybird procedures for solving linear systems. *Numerische Mathematik*, 67:1–19, 1994.

[8] C. Brezinski, M. R. Zaglia, and H. Sadok. Avoiding breakdown and near-breakdown in Lanczos type algorithms. *Numerical Algorithms*, 1:261–284, 1991.

[9] C. Brezinski, M. R. Zaglia, and H. Sadok. A Breakdown-free Lanczos type algorithm for solving linear systems. *Numerische Mathematik*, 63:29–38, 1992.

[10] C. Brezinski, M. R. Zaglia, and H. Sadok. New look-ahead Lanczos-type algorithms for linear systems. *Numerische Mathematik*, 83:53–85, 1999.

[11] C. Brezinski, M. R. Zaglia, and H. Sadok. The matrix and polynomial approaches to Lanczos-type algorithms. *Journal of Computational and Applied Mathematics*, 123:241–260, 2000.

[12] C. Brezinski, M. R. Zaglia, and H. Sadok. A review of formal orthogonality in Lanczos-based methods. *Journal of Computational and Applied Mathematics*, 140:81–98, 2002.

[13] C. G. Broyden and M. T. Vespucci. *Krylov Solvers For Linear Algebraic Systems*. Elsevier, Amsterdam, The Netherlands, 2004.

[14] D. Calvetti, L. Reichel, F. Sgallari, and G. Spaletta. A Regularizing Lanczos iteration method for underdetermined linear systems. *Jouranl of Computational and Applied Mathematics*, 115:101–120, 2000.

[15] G. Cybenko. An explicit formula for Lanczos plonomials. *Linear Algebra Appl.*, 88/89:99–115, 1987.

[16] A. Draux. *Polynômes Orthogonaux Formels. Application, LNM 974*. Springer-Verlag, Berlin, 1983.

[17] M. Farooq. *New Lanczos-type Algorithms and their Implementation*. PhD thesis, University of Essex, UK, 2011. `http://serlib0.essex.ac.uk/record=b1754556`.

[18] M. Farooq and A. Salhi. New Recurrence Relationships between Orthogonal Polynomials which Lead to New Lanczos-type Algorithms. *Journal of Prime Research in Mathematics*, 8:61–75, 2012.

[19] M. Farooq and A. Salhi. A Restarting Approach to Beating the Inherent Instability of Lanczos-type Algorithms. *Iranian Journal of Science and Technology, Transaction A-Science*, 37(3.1):349–358, 2013.

[20] M. Farooq and A. Salhi. A Switching Approach to Avoid Breakdown in Lanczos-type Algorithms. *Applied Mathematics and Information Sciences*, 8(5):2161–2169, 2014.

[21] M. Farooq and A. Salhi. A new Lanczos-type algorithm for system of linear equations. *Journal of Prime Research in Mathematics*, 10:116–121, 2015.

[22] R. Fletcher. Conjugate Gradient methods for indefinite systems. In G.A. Watson, editor, *Numerical Analysis, Dundee 1975, Lecture Notes in Mathematics,*, volume 506. Springer, Berlin, 1976.

[23] A. Greenbaum. *Iterative Methods for Solving Linear System*. Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[24] A. El Guennouni. A unified approach to some strategies for the treatment of breakdown in Lanczos-type algorithms. *Applicationes Mathematicae*, 26:477–488, 1999.

[25] M. R. Hestenes and E. Stiefel. Mehtods of Conjugate Gradients for solving linear systems. *Journal of the National Bureau of Standards*, 49:409–436, 1952.

[26] C. Lanczos. An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integeral Operators. *Journal of Research of the National Bureau of Standards*, 45:255–282, 1950.

[27] C. Lanczos. Solution of systems of linear equations by minimized iteration. *Journal of the National Bureau of Standards*, 49:33–53, 1952.

[28] G. Meurant. *The Lanczos and Conjugate Gradient algorithms, From Theory to Finite Precision Computations*. SIAM, Philadelphia, **2006**.

[29] B. N. Parlett and D. S. Scott. The Lanczos Algorithm With Selective Orthogonaliztion. *Mathematics of Computation*, 33:217–238, 1979.

[30] B. N. Parlett, D. R. Taylor, and Z. A. Liu. A Look-Ahead Lanczos Algorithm for Unsymmetric Matrices. *Mathematics of Computation*, 44:105–124, 1985.

[31] Y. Saad. On the Lanczos method for solving linear system with several right-hand sides. *Mathematics of Computation*, 48:651–662, 1987.

[32] G. Szegö. *Orthogonal Polynomials.* American Mathematical Society, Providence, Rhode Island, 1939.

[33] S. Ullah, M. Farooq, and A. Salhi. An alternative derivation of a new Lanczos-type algorithm for systems of linear equations. *Punjab University Journal of Mathematics*, 45:39–49, 2013.

[34] H. A. Van Der Vorst. An iterative solution method for solving f$(A)\mathbf{x}$=$\mathbf{b}$, using Krylov subspace information obtained for the symmetric positive definite matrix $A$. *Journal of Computational and Applied Mathematics*, 18(2):249–263, 1987.

[35] Q. Ye. A Breakdown-Free Variation of the Nonsymmetric Lanczos Algorithms. *Mathematics of Computation*, 62:179–207, 1994.