# New Remedial Approaches to the Breakdown of Lanczos-type Algorithms

**Syed Muhammad Ghufran**

A Thesis presented for the degree of

**Doctor of Philosophy**

at the

Department of Mathematical Sciences

University of Essex

October 2017

*Dedicated to*

In loving memory of my Father and Mother, my so sweet children and all who continually pray for my fortune.

*SYED ABDUL GHAFFAR (Late)*

*SHEREEN TAJ*

# Acknowledgements

I would like to take this opportunity to acknowledge many people who have assisted me during my studies.

First of all, I gratefully acknowledge the support and generosity of my supervisor Professor Abdellah Salhi, his patience, time and most important his guidence and support in this research.

To Abdul Wali Khan University, Mardan, Khyber Pakhtunkhwa, Pakistan for their financial support.

To my Supervisory Board member Dr Hongsheng Dai and Dr Hadi Susanto, who gave me their valuable suggestions and inputs. To the administrative staff specially Mrs Shauna Meyers, of Department of Mathematical Sciences, University of Essex for their kindness and assistance. To Dr Georgi Grahovski and Professor Edward Codling for giving me chance to work as GTA in their module and their invaluable guidance. To my office fellows at University of Essex for their help and guidance, specially Maharani, Arief Setyanto, Muhammad Sulaiman, Zardad Khan, Tanhi, Junaid Mustafa, Amal Mohammed, Omar Karakchi, Ms Nosheen and Muhammad Farooq.

To all my friends at Essex, with whom I have enjoyed my stay at Essex. Especially to Mr Abdulkareem Alzahrani, Mr Muhammad Ahmed Khan, Mr Ahmad Alzahrani, Mr

Zia ul Hassan Khan, Mr Atal Khan Gardiwal, Mr Javed, Muhammad Parvez, Mr Alhaji Bukar, Farrukh Rasheed and Abdul Aziz Aljohar who helped me a lot during hard times. I would also like to thank all my friends in Pakistan especially Mr Shah Jehan who always remember me in their prayer and pray for my success in research study.

Finally, I especially thank to my beloved children Laiqa Mehreen, Farwah Ghufran, Syed Muuamad Muneeb, Syed Muhammad Mutahir and my wife for their continuous support and patience. To my mother for their unconditional love, support and encouragement. To my brothers Syed Muhammad Nawaz, Musharraf Shah, Farhan Danish, and my sisters for their moral support, prayer and encouragement. To my family of brothers in laws, my uncles and my aunts for their encouragement, prayer and moral support throughout my studies.

# Abstract

There are numerous algorithms for the solution of systems of linear equations and eigenvalue problems. Among such methods, one of the best known iterative schemes is the Lanczos algorithm. It has however, a very serious shortcoming in that it break down frequently before achieving convergence to an acceptable solution. This project focuses on investigating this breakdown issue. There are a number of attempts to address it. Restarting and Switching as implemented previously by Farooq and Maharani, which rely on guessing the appropriate number of iterations before halting the Lanczos process and restarting it or switching to a different one. This guess is very sensitive to the type of problem solved, its data and size. If underestimated then the process is stopped too early, too often. This means that a lot of stable iterations are wasted, potentially. If, on the other hand, this number is over-estimated, then the process will breakdown which means that restarting and / or switching will be more costly. The aim of this thesis is to avoid guessing the number of iteration by monitoring the parameters of the recurrence relations on which the given Lanczos-type algorithms are based, which cause breakdown. This monitoring is targeted to the appropriate or problematic parameters. In this thesis we show that this approach is effective as it does not require too much extra work. At the same time it cuts on the wasted iterations and the full blown breakdown caused by inaccurate guesses of the number of iterations one has to let the algorithm run before halting it.

Although this is the core of our contributions in this thesis, we have also suggested new Lanczos-type algorithms and tested them against existing ones. This work complements that of Farooq, Mahrani, Baheux and the Brezinski team. The results show that we have made Lanczos-type algorithms old and new more reliable and robust.

# Declaration

The work in this thesis is based on research carried out at Department of Mathematical Sciences, University of Essex, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification, and it is all my own work, unless referenced, to the contrary, in the text.

**Copyright © 2017 by Syed Muhammad Ghufran**.

"The copyright of this thesis rests with the author. No quotations from it should be published without the author's prior written consent, and information derived from it should be acknowledged.".

# Contents

# List of Tables

# Chapter 1

# Introduction and Literature Review

## 1.1 Introduction

One of the most important tasks in numerical methods is the ability to solve the linear system

$$Ax = \mathbf{b}, \tag{1.1}$$

where $A \in \mathbb{R}^{n \times n}$ and $\mathbf{x}$, $\mathbf{b} \in \mathbb{R}^n$.

Systems of Linear Equations (SLEs) are an important practical problem in many aspects of life. It has found its way into natural sciences and management sciences. Therefore, solutions to this problem have to be found frequently. This means that new improvements, however small, are always welcome.

One way to solve it is to put it in matrix form and then use special techniques based on matrix algebra. For a small number of linear equations, the standard approach is to use direct methods [27,28], but for large and practical problems iterative methods are usually the norm [1,41,61,64,69].

In 1950, Cornelius Lanczos, introduced his algorithm [52]. The most prominent feature of the method is that it reduces a symmetric matrix $A$ into an equivalent tridiagonal one and initially it was aimed at finding eigenvalues and corresponding eigenvectors of matrix $A$ [59]. As the computation of eigenvalues of a matrix and the solution of the SLEs are equivalent problems, the Lanczos method for the eigenvalue problem was extended by Lanczos in 1952, to solve SLEs especially when they are large and sparse [53]. The Lanczos approach for solving (1.1), is an orthogonal projection method on Krylov subspace $\mathcal{K}_k(A, r_0)$ of order $k$ [64,65]. The definition of this space will be given in the next section. In the same year, 1952, another iterative scheme for solving SLEs was presented by Hestenes and Stiefel in [37,45], known as the Conjugate Gradient (CG) method. This method is useful when the matrix is symmetric and positive definite. In 1964, Lanczos and Householder pointed out that both the Lanczos and CG-method were the same for symmetric and positive definite matrices. Extension to the non-symmetric case was studied by Hestenes in [44]. In early periods, the Lanczos process was ignored by numerical analysts due to various reasons. One of the main ones is the loss of orthogonality in Lanczos vectors [53] which affects the accuracy in the iterative process as the accuracy of the Lanczos process is related to the orthogonality of Lanczos vectors.

In the last few decades, different variants of Lanczos algorithm have been designed. A transpose free algorithm was presented by C. Brezinski in [7]. In 1992, a Breakdown-free Lanczos-type algorithm was given in [17] which was known as MRZ (Method of Recursive Zoom). In [20] Brezinski derived new Lanczos algorithms using two different ways that are matrix and polynomial approaches. New variants of these algorithms have been derived by Baheux in [4] using recurrence relationships between Formal Orthogonal Polynomials

(FOPs) [5]. Recently in [33], Lanczos-type algorithms have been presented using new recurrence relationships between these FOPs. The Lanczos [53] method solves SLEs with an iterative process which gives the exact solution in a finite number of steps not greater than the dimension of the system, in exact arithmetic.

In the last few decades, different variants of Lanczos-type algorithms have been designed [4,20,23,33,38,42,48–50,60,63,66,68]. One particular weakness of the Lanczos-type algorithm is that, it easily breaks down, causing the process to stop. This is either due to a division by zero when computing the coefficients of those relations or due to the non-existence of FOP [12,18]. Division by a quantity close to zero causes *near-breakdown* thus producing numerical instability in the algorithm. These breakdown problems were partially solved in a series of papers by C. Brezinski, M. Redivo-Zaglia and H. Saddok, [7,13,15–17,19,22,25], and Farooq [33] and Maharani [55].

## 1.2 Objective and Approach of the Project

In this thesis our focus is mainly on the breakdown issues of Lanczos-type algorithms when solving large sparse systems of linear equations. The strategy adopted for avoiding the breakdown problem is monitoring the behaviour of the denominators and the components of the offending components of some of the coefficients involved in the recurrence relations that make up the Lanczos-type algorithm. We choose a threshold value $\epsilon$ for that component. When this component falls below $\epsilon$, for instance, $|c(x^k P_k)| \leq \epsilon$, where $c$ is linear functional, $P_k$ is the family of formal orthogonal polynomials and $x^i$ is a monic polynomial of degree $i$, then the process is stopped explicitly instead of letting it breakdown. We then restart it as fast as we can avoid wasting time due to recovering and resetting the process.

## 1.3 Thesis Outline

The thesis is organized as follows.

In Chapter 1 we briefly review the notion of Formal Orthogonal Polynomials. We discuss the basic theory of Lanczos-type algorithms for solving SLEs. The breakdown issue and the existing strategies to cure it are also explained.

In Chapter 2 we will extend the existing Lanczos-type algorithm using recurrence relationships between higher degree FOPs.

In Chapter 3 we will derive other variants of the Lanczos-type algorithm involving the ordinary polynomial $U_i(x) = P_i(x)$ and the monic polynomial $U_i(x) = P_i^{(1)}(x)$ instead of the standard auxiliary polynomial $U_i(x) = x^i$ that is used in Baheux [4] and Farooq [33]. The $P_i^{(1)}(x)$ in this selection is a monic polynomial of degree $i$ belonging to the family of FOPs with respect to the linear functional $c^{(1)}$ defined by $c^{(1)}(x^i) = c(x^{i+1})$.

In Chapter 4 we mainly discuss the prominent issues of breakdown in the Lanczos-type algorithms. We regularly monitor the components of those coefficients with denominators that blow up prior to breakdown. We suggest a stopping test that detects the imminence of a breakdown. It is used in restarting and switching strategies, that we are putting forward and implementing.

In Chapter 5 we suggest an alternative way to continue the solution process after it has been halted. This is the switching approach between different algorithms.

Chapter 6 contains conclusions and suggestions for further work.

## 1.4 Review of Literature

A number of concepts are needed for this study which include

- Understanding how the derivation of the Lanczos algorithm using the Krylov subspace method and its use in solving SLEs;
- The theory of Formal Orthogonal Polynomials (FOPs);
- The breakdown in the Lanczos-type algorithms and its remedies.

### 1.4.1 The Krylov Subspace Method (KSM)

Krylov subspace methods are widely used for solving a system of linear equations and eigenvalue problems, involving large and sparse matrices. They are popular iterative methods.

**Definition 1.4.1** Given $A \in \mathbb{R}^{n\times n}$ and $\mathbf{b} \in \mathbb{R}^n$ with $\mathbf{b} \neq 0$ then,

1. the *Krylov sequence* is
$$\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, A^3\mathbf{b}, \ldots,$$
2. the $k^{th}$ *Krylov Matrix* is
$$\mathrm{K_k} = [\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, \ldots, A^{k-1}\mathbf{b}],$$
3. the *Krylov subspace* of dimension $k$ is
$$\mathcal{K}_k(A, \mathbf{b}) = span\{\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, \ldots, A^{k-1}\mathbf{b}\}. \tag{1.2}$$

### 1.4.2 KSM for Solving SLEs

The Krylov subspace method for solving SLEs is given in [62,64,69]. Mathematically, KSMs are based on projection methods.

Consider (1.1) again. KSM is an iterative method stating with

- an initial approximation $\mathbf{x}_0$ to the solution of (1.1),
- an initial residual $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$.

The Krylov subspace of dimension $k$ defined by $A$ and $\mathbf{r}_0$ is

$$\mathcal{K}_k(A, \mathbf{r}_0) = span\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, ..., A^{k-1}\mathbf{r}_0\}.$$

Let $\mathcal{L}_k$ and $\mathcal{K}_k$ be the two subspaces of dimensions $k$. The idea behind KSM [54, 64] is solving the system (1.1) by choosing an initial approximate solution $\mathbf{x}_0$ and generating a sequence of approximate solutions $\mathbf{x}_k$ from

$$\mathbf{x}_0 + \mathcal{K}_k, \quad \text{and} \tag{1.3}$$

$$\mathbf{r}_k = (b - A\mathbf{x}_k) \perp \mathcal{L}_k, \tag{1.4}$$

is projection method is called the Krylov subspace method [6, 64]. Furthermore, according to the choice of $\mathcal{L}_k$ there exist several KSM [21]. For example, if $\mathcal{L}_k = \mathcal{K}_k(A^T, \mathbf{y})$, where $\mathbf{y}$ is some nonzero vector, then the KSM is known as the Lanczos method.

### 1.4.3 Formal Orthogonal Polynomials

Let $c_0, c_1$,... be a sequence of real and complex numbers. We define the *linear functional* $c$ on the vector space of complex polynomials by

$$c(x^i) = c_i, \quad i \geq 0, \tag{1.5}$$

The numbers $c_i$ are called the *moments* of $c$ [8].

**Definition 1.4.1** *The polynomials* $\{P_k\}$ *are said to form the family of Formal Orthogonal Polynomials [5,8,11] with respect to c if,* $\forall$ *k they are defined by*

1. $P_k$ has exact degree $k$,

2. $c(U_i(x)P_k(x)) = 0$ for $i = 0, ..., k-1$,

3. $c(U_i(x)P_k(x)) \neq 0$,

where $U_i(x)$ is the unitary polynomial of exact degree $i$ [4]. The second condition is called the *orthogonality condition*. Some of the choices of $U_i(x)$ are

- $U_i(x) = x^i$,
- $U_i(x) = P_i(x)$,
- $U_i(x) = P_i^{(1)}(x)$.

By linear combination, it can also be written as

$$c(p_i(x)P_k(x)) = 0 \quad \text{for} \quad i = 0, ..., k-1, \tag{1.6}$$

where $p_i(x)$ is any polynomial of degree $k-1$ at most. Thus, it also follows that

$$c(P_n(x)P_k(x)) = 0 \quad \text{for} \quad n \neq k, \tag{1.7}$$

when assumed that the degrees of both polynomials are different. If we set $P_k$ to be the polynomial assumed to exist as

$$P_k(x) = a_0 + a_1x + a_2x^2 + ... + a_kx^k, \tag{1.8}$$

and satisfying the orthogonality conditions which are equivalent to

$$c(x^iP_k(x)) = 0, \quad for \;\; i = 0, 1, 2, ..., k-1, \tag{1.9}$$

then

$$a_0c_i + a_1c_{i+1} + ... + a_kc_{i+k} = 0.$$

This is a system of $k$ equations in $k+1$ unknowns, of the form, for $i = 0, 1, ..., k-1$,

$$\begin{cases} a_0c_0 + a_1c_1 + ... + a_kc_k = 0, \\ a_0c_1 + a_1c_2 + ... + a_kc_{k+1} = 0, \\ \vdots \\ a_0c_{k-1} + a_1c_k + ... + a_kc_{2k-1} = 0. \end{cases} \tag{1.10}$$

Its solution is completely determined, once a supplementary condition has been added. Now adding an equation $-P_k(x) + a_0 + a_1x + a_2x^2 + ... + a_kx^k = 0$ to the system, we have $(k+1) \times (k+1)$ system of linear equations in $a_i$ for $i = 0, 1, ..., k$. The polynomial $P_k$ can be expressed by the determinantal formula as following [8, 9].

$$P_k(x) = \frac{1}{H_k^{(0)}} \begin{vmatrix} 1 & x & \cdots & x^k \\ c_0 & c_1 & \cdots & c_k \\ \vdots & \vdots & & \vdots \\ c_{k-1} & c_k & \cdots & c_{2k-1} \end{vmatrix}, \quad H_k^{(0)} = \begin{vmatrix} c_1 & \cdots & c_k \\ \vdots & & \vdots \\ c_k & \cdots & c_{2k-1} \end{vmatrix}. \tag{1.11}$$

Where the denominator of $P_k(x)$ is the *Hankel determinant* $H_k^{(0)}$ [23]. It is clear that $P_k(x)$ exists if and only if $H_k^{(0)} \neq 0$. The normalization of $P_k(x)$ is obtained by the condition $P_k(0) = 1$. If for some $k$, $H_k^{(0)} = 0$, then $P_k$ does not exist, and the breakdown occurs in the solution process.

### 1.4.4 Adjacents Families of FOP

We consider the linear functionals $c^{(n)},\ n = 0, 1, ...,$ defined by

$$c^{(n)}(x^i) = c(x^{n+i}) = c_{n+i}, \quad i = 0, 1, ..., \tag{1.12}$$

with the assumption that $c_i = 0$ if $i < 0$, [8].

Let us consider $P_k^{(n)}$ be the family of monic FOP's with respect to $c^{(n)}$, such that

$$c^{(n)}(x^i P_k^{(n)}(x)) = 0, \quad i = 0, 1, ..., k-1. \tag{1.13}$$

Thus the polynomials $P_k^{(0)}$ are identical to the polynomials $P_k$ defined above. $P_k^{(1)}$ is the family of *monic* formal orthogonal polynomials of degree $k$ (where $a_k$ is the coefficient of $x^k$ in $P_k^{(1)}$ equal to 1), with respect to a linear functional $c^{(1)}$ defined by

$$c^{(1)}(x^i) = c(x^{i+1}) = c_{i+1}, \quad i = 0, 1, ..., \tag{1.14}$$

and which satisfies the orthogonality conditions

$$c^{(1)}(x^i P_k^{(1)}(x)) = c(x^{(i+1)} P_k) = 0, \quad i = 0, 1, ..., k-1. \tag{1.15}$$

Now consider the monic polynomials $P_k^{(1)}(x)$ defined by the determinantal formula, [23,34].

$$P_k^{(1)}(x) = \frac{\begin{vmatrix} c_1 & c_2 & \cdots & c_{k+1} \\ \vdots & \vdots & & \vdots \\ c_k & c_{k+1} & \cdots & c_{2k} \\ 1 & x & \cdots & x^k \end{vmatrix}}{H_k^{(0)}}. \tag{1.16}$$

$P_k^{(1)}(x)$ exists if and only if $H_k^{(0)} \neq 0$, hence $P_k(x)$ and $P_k^{(1)}(x)$ exist under the same condition. So $\{P_k\}$ and$\{P_k^{(1)}\}$ are called *adjacent families* of FOPs [8, 9]. There exist many recurrence relations between the two adjacent families of polynomials $P_k$ and $P_k^{(1)}$ [3, 4, 15, 17]. More relations have been studied in [33], leading to new Lanczos-type algorithms.

## 1.5 The Lanczos Approach

Let us consider a linear system of equations (1.1) again For solving this system, the Lanczos method [51–53, 56] consists in constructing a sequence of vectors $\mathbf{x}_k \in R^n$ defined by the following steps, [21]:

1. choose two arbitrary vectors $\mathbf{x}_0$ and $\mathbf{y}$ in $R^n$ such that $\mathbf{y} \neq 0$,
2. set $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$,
3. determine $\mathbf{x}_k$ such that

$$\mathbf{x}_k - \mathbf{x}_0 \in \mathcal{K}_k(A, \mathbf{r}_0) = span\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, ..., A^{k-1}\mathbf{r}_0\}, \tag{1.17}$$

$$\mathbf{r}_k = b - A\mathbf{x}_k \perp \mathcal{K}_k(A^T, \mathbf{y}) = span\{\mathbf{y}, A^T\mathbf{y}, (A^t)^2\mathbf{y}, ..., (A^T)^{k-1}\mathbf{y}\}, \tag{1.18}$$

where $\mathcal{K}_k(A, \mathbf{r}_0)$ is called a *Krylov* subspace and $A^T$ is the transpose of $A$.

From eq (1.17), we set $\mathbf{x}_k - \mathbf{x}_0$ as

$$\mathbf{x}_k - \mathbf{x}_0 = -a_1\mathbf{r}_0 - a_2A\mathbf{r}_0 - a_3A^2\mathbf{r}_0 - ... - a_kA^{k-1}\mathbf{r}_0.$$

Now, multiplying both sides by $A$ and adding and subtracting $\mathbf{b}$ on the left hand side, we obtain

$$\mathbf{r}_k = \mathbf{r}_0 + a_1A\mathbf{r}_0 + a_2A^2\mathbf{r}_0 + ... + +a_kA^k\mathbf{r}_0. \tag{1.19}$$

From (1.18), the orthogonality condition gives

$$(A^{T^i}y, \mathbf{r}_k) = (y, A^i\mathbf{r}_k) = (y, A^iP_k(A)\mathbf{r}_0) = 0, \text{ for } i = 0.1, ..., k-1.$$

By (1.19)

$$(\mathbf{y}, A^i\mathbf{r}_0 + a_1A^{i+1}\mathbf{r}_0 + a_2A^{i+2}\mathbf{r}_0 + ... + a_kA^{i+k}\mathbf{r}_0) = 0,$$

$$(\mathbf{y}, A^i\mathbf{r}_0) + a_1(y, A^{i+1}\mathbf{r}_0) + ... + a_k(y, A^{i+k}\mathbf{r}_0) = 0,$$

we obtain the following system of linear equations

$$\begin{cases} a_1(\mathbf{y}, A\mathbf{r}_0) + ... + a_k(\mathbf{y}, A^k\mathbf{r}_0) = -(\mathbf{y}, \mathbf{r}_0), \\ a_1(A^T\mathbf{y}, A\mathbf{r}_0) + ... + a_k(A^T\mathbf{y}, A^k\mathbf{r}_0) = -(A^T\mathbf{y}, \mathbf{r}_0), \\ \vdots \\ a_1((A^T)^{k-1}\mathbf{y}, A\mathbf{r}_0) + ... + a_k((A^T)^{k-1}\mathbf{y}, A^k\mathbf{r}_0) = -((A^T)^{k-1}\mathbf{y}, \mathbf{r}_0). \end{cases} \tag{1.20}$$

If the determinant of (1.20) is different from zero then its solution exists and formulae (1.17) and (1.18) allow to obtain $\mathbf{x}_k$ and $\mathbf{r}_k$. Obviously, solving systems (1.20) is impractical. Such computation is feasible as the polynomials $P_k$ form a family of FOPs, with respect to the linear functional $c$ [10,71]. The easiest way to get the solutions of the system is by computing recursively the polynomial $P_k(x)$.

If we consider the polynomial

$$P_k(x) = 1 + a_1x + a_2x^2... + a_kx^k, \tag{1.21}$$

then $\mathbf{r}_k$ can be written as

$$\mathbf{r}_k = P_k(A)\mathbf{r}_0. \tag{1.22}$$

The polynomial $P_k$ is known as the residual polynomial [23]. Let $c$ be the linear functional [8] defined by

$$c(x^i) = c_i, \quad \text{for} \quad i \geq 0, \tag{1.23a}$$

Moreover by setting

$$c_i = (\mathbf{y}, A^i\mathbf{r}_0), \quad \text{for} \quad i = 0, 1, ..., \tag{1.23b}$$

then the system (1.20) can be written as

$$c_i + a_1c_{i+1} + ... + a_kc_{i+k} = 0, \quad \text{for} \quad i = 0, 1, ..., k-1.$$

The preceding orthogonality conditions are equivalent to

$$c(x^i P_k(x)) = 0, \quad \text{for } \; i = 0, 1, \ldots k-1. \tag{1.24}$$

These conditions show that $P_k$ is the polynomial of degree at most $k$ belonging to the formal orthogonal polynomials with respect to $c$, normalized by the condition $P_k(0) = 1$. Since the polynomial $P_k(x)$ in (1.21), can be written as

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

Replace $x$ by $A$ and also multiply both side by $\mathbf{r}_0$ in the last relation, to get

$$\mathbf{r}_k = \mathbf{r}_0 + AQ_{k-1}(A)\mathbf{r}_0, \tag{1.25}$$

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

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

and multiplying both sides by $-A^{-1}$, we get

$$\mathbf{x}_k = \mathbf{x}_0 - Q_{k-1}(A)\mathbf{r}_0. \tag{1.26}$$

Which shows that $\mathbf{x}_k$ can be computed from $\mathbf{r}_k$ without using $A^{-1}$. This is the Lanczos method.

## 1.6 Classification

There exist several recurrence relationships for implementing Lanczos methods. They can all be derived using the theory of FOPs. Here, we consider two families of FOPs $P_k(x)$ and $P_k^{(1)}(x)$. The polynomial $P_k(x)$ will be related to the residual $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$ of the Lanczos method by $\mathbf{r}_k = P_k(A)\mathbf{r}_0$, while the polynomial $P_k^{(1)}(x)$ will define $\mathbf{z}_k = P_k^{(1)}(A)\mathbf{r}_0$. They are represented by $A_i$ and $B_j$ for $P_k(x)$ and $P_k^{(1)}(x)$ respectively. The Lanczos-type algorithm based only on relations $A_i$ are named $A_i$-type algorithms, and those which are characterized by both types of the relations $A_i$ and $B_j$ are represented by $A_i/B_j$-type Lanczos algorithms. [4,23,34].

C. Baheux and C. Brezinski [3,4,23] studied the relations where the degrees of the polynomials in the right and left hand sides of the relation differ by one or two at most. In Farooq's work [33,34] the difference in degrees is two or three. We will adopt the same idea here and extend the list accordingly, where the difference of the degrees in the relations is three or four. They are given in Tables 1.1-1.3

**Table 1.1:** *Computation formulae of $A_i$ and $B_j$ from different polynomials [4].*

| Relation $A_i$ | Computation of $P_k$ from | | Relation $B_j$ | Computation of $P_k^{(1)}$ from | |
|---|---|---|---|---|---|
| $A_1$ | $P_{k-2}$ | $P_{k-2}^{(1)}$ | $B_1$ | $P_{k-2}$ | $P_{k-2}^{(1)}$ |
| $A_2$ | $P_{k-2}$ | $P_{k-1}^{(1)}$ | $B_2$ | $P_{k-2}$ | $P_{k-1}^{(1)}$ |
| $A_3$ | $P_{k-2}$ | $P_k^{(1)}$ | $B_3$ | $P_{k-2}$ | $P_k$ |
| $A_4$ | $P_{k-2}$ | $P_{k-1}$ | $B_4$ | $P_{k-2}$ | $P_{k-1}$ |
| $A_5$ | $P_{k-2}^{(1)}$ | $P_{k-1}$ | $B_5$ | $P_{k-2}^{(1)}$ | $P_{k-1}$ |
| $A_6$ | $P_{k-2}^{(1)}$ | $P_{k-1}^{(1)}$ | $B_6$ | $P_{k-2}^{(1)}$ | $P_{k-1}^{(1)}$ |
| $A_7$ | $P_{k-2}^{(1)}$ | $P_k^{(1)}$ | $B_7$ | $P_{k-2}^{(1)}$ | $P_k$ |
| $A_8$ | $P_{k-1}^{(1)}$ | $P_{k-1}$ | $B_8$ | $P_{k-1}^{(1)}$ | $P_{k-1}$ |
| $A_9$ | $P_{k-1}$ | $P_k^{(1)}$ | $B_9$ | $P_{k-1}$ | $P_k$ |
| $A_{10}$ | $P_{k-1}^{(1)}$ | $P_k^{(1)}$ | $B_{10}$ | $P_{k-1}^{(1)}$ | $P_k$ |

**Table 1.2:** *Computation formulae of $A_i$ and $B_j$ from different polynomials [33].*

| Relation $A_i$ | Computation of $P_k$ from | | Relation $B_j$ | Computation of $P_k^{(1)}$ from | |
|---|---|---|---|---|---|
| $A_{11}$ | $P_{k-3}$ | $P_{k-1}^{(1)}$ | $B_{11}$ | $P_{k-3}$ | $P_{k-1}$ |
| $A_{12}$ | $P_{k-2}$ | $P_{k-3}$ | $B_{12}$ | $P_{k-2}$ | $P_{k-3}$ |
| $A_{13}$ | $P_{k-2}$ | $P_{k-3}^{(1)}$ | $B_{13}$ | $P_{k-2}^{(1)}$ | $P_{k-3}^{(1)}$ |
| $A_{14}$ | $P_{k-2}^{(1)}$ | $P_{k-3}^{(1)}$ | $B_{14}$ | $P_{k-3}^{(1)}$ | $P_{k-1}$ |
| $A_{15}$ | $P_{k-3}^{(1)}$ | $P_{k-1}^{(1)}$ | $B_{15}$ | $P_{k-2}$ | $P_{k-2}^{(1)}$ |
| $A_{16}$ | $P_{k-2}$ | $P_{k-2}^{(1)}$ | $B_{16}$ | $P_{k-2}^{(1)}$ | $P_{k-1}$ |
| $A_{17}$ | $P_{k-2}$ | $P_{k-1}^{(1)}$ | - | - | - |
| $A_{18}$ | $P_{k-1}^{(1)}$ | $P_{k-2}^{(1)}$ | - | - | - |
| $A_{19}$ | $P_{k-2}^{(1)}$ | $P_{k-1}$ | - | - | - |

**Table 1.3:** *Computation formulae of $A_i$ and $B_j$ from different polynomials*

| Relation $A_i$ | Computation of $P_k$ from | | Relation $B_j$ | Computation of $P_k^{(1)}$ from | |
|---|---|---|---|---|---|
| $A_{20}$ | $P_{k-3}$ | $P_{k-4}$ | $B_{17}$ | $P_{k-4}$ | $P_{k-2}$ |
| $A_{21}$ | $P_{k-4}$ | $P_{k-2}^{(1)}$ | $B_{18}$ | $P_{k-3}$ | $P_{k-4}$ |
| $A_{22}$ | $P_{k-3}$ | $P_{k-4}^{(1)}$ | $B_{19}$ | $P_{k-3}^{(1)}$ | $P_{k-4}^{(1)}$ |
| $A_{23}$ | $P_{k-3}^{(1)}$ | $P_{k-4}^{(1)}$ | $B_{20}$ | $P_{k-4}^{(1)}$ | $P_{k-2}$ |
| $A_{24}$ | $P_{k-4}^{(1)}$ | $P_{k-2}^{(1)}$ | $B_{21}$ | $P_{k-3}$ | $P_{k-3}^{(1)}$ |
| $A_{25}$ | $P_{k-3}$ | $P_{k-3}^{(1)}$ | - | - | - |
| $A_{26}$ | $P_{k-3}$ | $P_{k-2}^{(1)}$ | - | - | - |
| $A_{27}$ | $P_{k-2}^{(1)}$ | $P_{k-3}^{(1)}$ | - | - | - |
| $A_{28}$ | $P_{k-3}^{(1)}$ | $P_{k-2}$ | - | - | - |

## 1.7 The Breakdown Issue in Lanczos-type Algorithms

The Lanczos-type algorithms for solving systems of linear equations are based on formal orthogonal polynomials. Different variants of Lanczos-type algorithms have been derived using recurrence relationships between polynomials of a family of orthogonal polynomials or between those adjacent to families of orthogonal polynomials. When computing the coefficients of the FOPs involved in these recurrence relationships, which are in the ratio of scalar products, and these scalar products in the denominator become zero, then breakdown occurs in the algorithm. When such a scalar product is nearly equal to zero *(near-breakdown)* [14,15,18] then rounding errors can seriously affect the numerical stability of the algorithm and the process has to be stopped [13, 14, 18]. To illustrate the breakdown condition in calculating the recurrence relationships, let us consider the three-term recurrence relationship of a monic polynomial $P_{k+1}(x)$ as follows [12].

$$P_{k+1}(x) = (A_{k+1}x + B_{k+1})P_k - C_{k+1}P_{k-1}, \tag{1.27}$$

for $k = 0, 1, 2, ...,$ with $P_{-1}(x) = 0$ and $P_0(x) = 1$, where the coefficients $A_{k+1}$, $B_{k+1}$ and $C_{k+1}$ appearing in the relations are obtained by imposing the orthogonality condition with respect to the linear function $c$ on both sides. This leads to

$$c(x^i P_{k+1}) = A_{k+1}c(x^{i+1}P_k(x)) + B_{k+1}c(x^i P_k) + C_{k+1}c(x^i P_{k-1}),$$

$$A_{k+1}c(x^{i+1}P_k(x)) + B_{k+1}c(x^i P_k) + C_{k+1}c(x^i P_{k-1}) = 0.$$

The orthogonality condition is always true for $i = 0, 1, 2, ..., k$. Therefore for $i = k - 1$,

$$A_{k+1}c(x^k P_k(x)) - C_{k+1}c(x^{k-1}P_{k-1}) = 0. \tag{1.28}$$

For $i = k$,

$$A_{k+1}c(x^{k+1}P_k(x)) + B_{k+1}c(x^x P_k(x) - C_{k+1}c(x^k P_{k-1}(x)) = 0. \tag{1.29}$$

The normalization conditions $P_{k+1}(0) = 1$ give the third equation

$$B_{k+1} - C_{k+1} = 1. \tag{1.30}$$

So, we obtain a system of three equations for three unknowns $A_{k+1}$,$B_{k+1}$ and $C_{k+1}$. The determinant of the above $3 \times 3$ system of linear equations is given by

$$\Delta_k = -c(x^k P_k)[c(x^k P_k) - c(x^k P_{k-1})] - c(x^{k-1}P_{k-1})c(x^{k+1}P_k). \tag{1.31}$$

This system may be singular ($\Delta_k = 0$) and a breakdown can occur in the recurrence relationship even if $P_{k+1}$ ($H^{(1)}_{k+1} \neq 0$) exists and so, the recurrence relation cannot be used. This kind of breakdown is called *ghost breakdown* [18], which occurs due to the relation used for its computation. It does not correspond to the non-existence of an orthogonal polynomial of the family. When a breakdown occurs for some value of $k$, if $H^{(1)}_{k+1}$=0 then the corresponding orthogonal polynomial $P_{k+1}$ does not exist and a breakdown is due to the nonexistence of the polynomial which is called a *true breakdown* [18].

Several procedures for that purpose are present in the literature in the last few decades.

These breakdown problems were partially solved in a series of papers by C. Brezinski, M. Redivo-Zaglia and H. Saddok, [12,13,15,17,19,43] and Farooq [33] and Maharani [55]. There are many possible strategies to cure a breakdown issues in the Lanczos-type algorithms.

Breakdown can be avoided by jumping over the polynomials involved or over those that cannot be computed by the recurrence relationship under consideration, [15, 17]. In this case, more complicated recurrences based on those given in [19] have to be used.

The problem of near-breakdown, due to a division by scalar product close to zero, can be treated in a similar way as in [19]. The theory of Formal Orthogonal Polynomials greatly simplifies the treatment of breakdowns and near-breakdowns as shown in [20,30,31]. Other strategies such as restarting the Lanczos-type algorithms and switching between them have also been considered [35, 36].

## 1.8 Remedial Strategies

As mentioned earlier, Lanczos-type algorithms suffer from breakdown. Several procedures for dealing with these breakdowns are present in the literature. Recently alternative ways implement restarting and switching between algorithms.

### 1.8.1 Restarting Strategies

Restarting of iterative methods to avoid breakdown and improve convergence is not new [57]. This is one way to avoid the breakdown in the Lanczos-type algorithms. This strategies consists of restarting the same algorithm that fails [24,33,35,58], when breakdown occurs in the algorithm due to the non-existence of some coefficients of the FOPs involved in its recurrence relations. In these strategies, the idea is either to stop the Lancozs-type algorithm pre-emptively and restart it with some iterate or wait until breakdown occurs and then restart from the last iterate found. It is reasonable to restart from the point immediately before the breakdown occurred if one can detect it. Otherwise, one may consider restarting strategy after breakdown has happened [36]. Different strategies can be used for restarting various algorithms. In this procedure the algorithm starts working in a different Krylov subspace than the one it started with. These strategies are listed below. Note that ST stands for " Strategy".

1. **Restarting After Breakdown**: In this strategy, a particular Lanczos algorithm is run until a breakdown occurs. After the breakdown, the same Lanczos algorithm is restarted, but this time initializing it with the last iterate of the previously failed algorithm. This strategy is named **ST1**.

2. **Pre-emptive Restarting**: In this strategy, a Lanczos-type algorithm is run iteratively. Then it is halted and restarted again initializing it with the last iterate. While doing so, it can not be guaranteed that a breakdown will not happen before the interval end. This strategy is named **ST2**.

3. **Breakdown Monitoring**: In this strategy, the coefficients with the denominators causing the breakdown are regularly examined. When the values of these coefficients become less than a specified threshold then switching to another algorithm is implemented. This strategy is named **ST3**.

### 1.8.2 Switching Strategies

Switching is another way of curing breakdown in Lanczos-type algorithms. It follows the same pattern as restarting. In the switching strategy, different methods can be followed between two or more algorithms. If the running algorithm is switched to another algorithm based on different recurrence relations then this will be a proper switching.

## 1.9 Summary

In this chapter we have discussed the basic Lanczos process for solving systems of linear equations, the theory of Formal Orthogonal Polynomials (FOP's) on which the Lanczos-type algorithms are based. We have also discussed the breakdown issue in these algorithms and the current procedures for curing it. A brief review of the relevant literature was also given. The next chapter will consider the design of Lanczos-type algorithms based on recurrence relationships between FOPs of higher degrees than previously considered. These relations are in Table 1.3. Then we will compare the experiemental results of the new algorithm with the existing algorithms in [4, 33].

# Chapter 2

# Recursive Computation Based on High Degree FOPs and Lanczos-type Algorithms

## 2.1 Introduction

In this chapter, we introduce Lanczos-type algorithms based on high degree FOPs. We will derive new recurrence relationships which will be used for the derivation of these new Lanczos-type algorithms, [23,33,67]. C. Brezinski and his colleagues discussed all the variations which are expressed in Chapter 1, [3,4,11,23,29,30,63]. We will follow the same notation.

## 2.2 Recursive Computation Between the FOPs for $A_i$

First, we will derive some relationships $A_i$ ($i > 19$) for $P_k$ which can be used to find $\mathbf{r}_k$ and then $\mathbf{x}_k$ without using $A^{-1}$. We will only find the coefficients of the recurrence relations by using the orthogonality condition (1.24), which can be used for the implementation of Lanczos-type algorithms. However, if a recurrence relation exists but cannot be used for the implementation of Lanczos algorithm then there is no need to calculate its coefficients. The reason for this will be given. If we consider the condition

$$c(U_i P_k) = 0, \qquad \forall \quad i = 0, 1, ...., k-1, \tag{2.1}$$

$$c^{(1)}(U_i P_k^{(1)}) = 0, \quad \forall \quad i = 0, 1, ...., k-1, \tag{2.2}$$

where $U_i$ be an arbitrary family of polynomials [4] of exact degree $i$, then some of the possible choices of $U_i(x)$ are

- $U_i(x) = x^i$,
- $U_i(x) = P_i(x)$,
- $U_i(x) = P_i^{(1)}(x)$.

### 2.2.1 $A_{20}$ for $U_i(x) = x^i$

Consider the following recurrence relationship for $k \geq 4$,

$$P_k(x) = A_k\Big\{(x^3 + B_k x^2 + C_k x + D_k)P_{k-3} + (E_k x^4 + F_k x^3 + G_k x^2 + H_k x + I_k)P_{k-4}\Big\}, \tag{2.3}$$

where $P_k(x)$, $P_{k-3}(x)$ and $P_{k-4}(x)$ are polynomials of degree $k$, $k-3$ and $k-4$ respectively. The constant coefficients $A_k$, $B_k$, $C_k$, $D_k$, $E_k$, $F_k$, $G_k$, $H_k$ and $I_k$ are determined by $P_k(0) = 1$ and imposing the orthogonality condition (2.1).

Since $P_k(0) = 1$, $\forall\ k$, then for $x = 0$, equation (2.3) becomes

$$A_k = \frac{1}{D_k + I_k}. \tag{2.4}$$

After multiplying (2.3) by $x^i$ and applying the linear functional $c$ on both sides it becomes

$$c(x^iP_k) = A_k\Big\{c(x^{i+3}P_{k-3}) + B_kc(x^{i+2}P_{k-3}) + C_kc(x^{i+1}P_{k-3}) + D_kc(x^iP_{k-3}) + E_kc(x^{i+4}P_{k-4})$$
$$+ F_kc(x^{i+3}P_{k-4}) + G_kc(x^{i+2}P_{k-4}) + H_kc(x^{i+1}P_{k-4}) + I_kc(x^iP_{k-4})\Big\}. \quad (2.5)$$

Consequently, by applying (2.1), we have the relation for $i = 0, 1, ..., k-1$,

$$c(x^{i+3}P_{k-3}) + B_kc(x^{i+2}P_{k-3}) + C_kc(x^{i+1}P_{k-3}) + D_kc(x^iP_{k-3}) + E_kc(x^{i+4}P_{k-4}) + F_kc(x^{i+3}P_{k-4})$$
$$+ G_kc(x^{i+2}P_{k-4}) + H_kc(x^{i+1}P_{k-4}) + I_kc(x^iP_{k-4}) = 0. \quad (2.6)$$

The orthogonality condition is always true for $i = 0, 1, 2, ......, k-9$.

Therefore for $i = k-8$, equation (2.6) gives

$$E_kc(x^{k-4}P_{k-4}) = 0 \quad \Rightarrow \quad c(x^{k-4}P_{k-4}) \neq 0, \quad E_k = 0.$$

For $i = k-7$, equation (2.6) gives

$$F_kc(x^{k-4}P_{k-4}) = 0 \;\Rightarrow\; c(x^{k-4}P_{k-4}) \neq 0,\; F_k = 0.$$

For $i = k-6$, equation (2.6) gives

$$G_k = -\frac{c(x^{k-3}P_{k-3})}{c(x^{k-4}P_{k-4})}. \quad (2.7)$$

For $i = k-5$, equation (2.6) gives

$$B_kc(x^{k-3}P_{k-3}) + H_kc(x^{k-4}P_{k-4}) = -c(x^{k-2}P_{k-3}) - G_kc(x^{k-3}P_{k-4}). \quad (2.8)$$

For $i = k-4$, equation (2.6) gives

$$B_kc(x^{k-2}P_{k-3}) + C_kc(x^{k-3}P_{k-3}) + H_kc(x^{k-3}P_{k-4}) + I_kc(x^{k-4}P_{k-4}) = -c(x^{k-1}P_{k-3}) - G_kc(x^{k-2}P_{k-4}). \quad (2.9)$$

For $i = k-3$, and equation (2.6) gives

$$B_kc(x^{k-1}P_{k-3}) + C_kc(x^{k-2}P_{k-3}) + D_kc(x^{k-3}P_{k-3}) + H_kc(x^{k-2}P_{k-4}) + I_kc(x^{k-3}P_{k-4})$$
$$= -c(x^kP_{k-3}) - G_kc(x^{k-1}P_{k-4}). \quad (2.10)$$

For $i = k-2$, and equation (2.6) gives

$$B_k c(x^k P_{k-3}) + C_k c(x^{k-1} P_{k-3}) + D_k c(x^{k-2} P_{k-3}) + H_k c(x^{k-1} P_{k-4} + I_k c(x^{k-2} P_{k-4}) = -c(x^{k+1} P_{k-3}) - G_k c(x^k P_{k-4}). \quad (2.11)$$

For $i = k-1$, and equation (2.6) gives

$$B_k c(x^{k+1} P_{k-3}) + C_k c(x^k P_{k-3}) + D_k c(x^{k-1} P_{k-3}) + H_k c(x^k P_{k-4} + I_k c(x^{k-1} P_{k-4}) = -c(x^{k+2} P_{k-3}) - G_k c(x^{k+1} P_{k-4}). \quad (2.12)$$

Equations (2.8), (2.9), (2.10), (2.11) and (2.12) can be written as

$$\begin{cases} a_{11} B_k + a_{14} H_k = b_1, \\ a_{21} B_k + a_{22} C_k + a_{24} H_k + a_{25} I_k = b_2, \\ a_{31} B_k + a_{32} C_k + a_{33} D_k + a_{34} H_k + a_{35} I_k = b_3, \\ a_{41} B_k + a_{42} C_k + a_{43} D_k + a_{44} H_k + a_{45} I_k = b_4, \\ a_{51} B_k + a_{52} C_k + a_{53} D_k + a_{54} H_k + a_{55} I_k = b_5. \end{cases} \quad (2.13)$$

Where $a_{11}, a_{14}, a_{21}, a_{22}, a_{24}, a_{25}, a_{31}, a_{32}, a_{33}, a_{34}, a_{35}, a_{41}, a_{42}, a_{43}, a_{44}, a_{45}, a_{51}, a_{52}, a_{53}, a_{54}$, and $a_{55}$ are the coefficients of $B_k$, $C_k$, $D_k$, $H_k$ and $I_k$ respectively. Suppose $b_1$, $b_2$, $b_3$, $b_4$, and $b_5$ are the corresponding right hand side terms of these equations. If $\Delta_k$ represents the determinant of the coefficients matrix of (2.13) then we have,

$$\Delta_k = det(V), \quad (2.14)$$

where $V = matrix([v_1, v_2, v_3, v_4, v_5])$,

$v_1 = [a_{11}, 0, 0, a_{14}, 0],\ v_2 = [a_{21}, a_{22}, 0, a_{24}, a_{25}],\ v_3 = [a_{31}, a_{32}, a_{33}, a_{34}, a_{35}]$,

$v_4 = [a_{41}, a_{42}, a_{43}, a_{44}, a_{45}],\ v_5 = [a_{51}, a_{52}, a_{53}, a_{54}, a_{55}]$.

If $\Delta_k \neq 0$, then

$$\begin{cases} B_k = \frac{det(W)}{\Delta_k}, \quad \text{where } W = matrix([w_1, w_2, w_3, w_4, w_5]), \\ w_1 = [b_1, 0, 0, a_{14}, 0],\ w_2 = [b_2, a_{22}, 0, a_{24}, a_{25}],\ w_3 = [b_3, a_{32}, a_{33}, a_{34}, a_{35}], \\ w_4 = [b_4, a_{42}, a_{43}, a_{44}, a_{45}],\ w_5 = [b_5, a_{52}, a_{53}, a_{54}, a_{55}], \\ C_k = \frac{det(U)}{\Delta_k}, \quad \text{where } U = matrix([u_1, u_2, u_3, u_4, u_5]), \\ u_1 = [a_{11}, b_1, 0, a_{14}, 0],\ u_2 = [a_{21}, b_2, 0, a_{24}, a_{25}],\ u_3 = [a_{31}, b_3, a_{33}, a_{34}, a_{35}], \\ u_4 = [a_{41}, b_4, a_{43}, a_{44}, a_{45}],\ u_5 = [a_{51}, b_5, a_{53}, a_{54}, a_{55}], \\ H_k = \frac{b_1 - a_{11}B_k}{a_{14}}, \\ I_k = \frac{b_2 - a_{21}B_k - a_{22}C_k - a_{24}H_k}{a_{25}}, \\ D_k = \frac{b_3 - a_{31}B_k - a_{32}C_k - a_{34}H_k - a_{35}I_k}{a_{33}}. \end{cases} \tag{2.15}$$

Since $E_k = F_k = 0$, relation $A_{20}$ becomes

$$P_k(x) = A_k\Big\{(x^3 + B_kx^2 + C_kx + D_k)P_{k-3}(x) + (G_kx^2 + H_kx + I_k)P_{k-4}(x)\Big\}. \tag{2.16}$$

Therefore, $A_{20}$ leads to a Lanczos-type algorithm.

### 2.2.2 $A_{21}$ for $U_i(x) = x^i$

Consider the following recurrence relationship for $k \geq 4$

$$P_k(x) = (A_kx^4 + B_kx^3 + C_kx^2 + D_kx + E_k)P_{k-4} + (F_kx^2 + G_kx + H_k)P^{(1)}_{k-2}, \tag{2.17}$$

where $P_k$, $P^{(1)}_{k-2}$ and $P_{k-4}$ are polynomials of degree $k$, $k-1$ and $k-4$ respectively. The constant coefficients $A_k$, $B_k$, $C_k$, $D_k$, $E_k$ $F_k$ $G_k$ and $H_k$ are determined by $P_k(0) = 1$ and imposing the orthogonality condition (2.1). Since $P_k(0) = 1$, $\forall\ k$, then for $x = 0$, equation (2.17) becomes

$$E_k + H_kP^{(1)}_{k-2}(0) = 1. \tag{2.18}$$

After multiplying equation (2.17) by $x^i$ and applying linear functional $c$ on both sides it becomes

$$c(x^iP_k) = A_kc(x^{i+4}P_{k-4}) + B_kc(x^{i+3}P_{k-4}) + C_kc(x^{i+2}P_{k-4}) + D_kc(x^{i+1}P_{k-4}) + E_kc(x^iP_{k-4})+$$
$$F_kc(x^{i+2}P^{(1)}_{k-2}) + G_kc(x^{i+1}P^{(1)}_{k-2}) + H_kc(x^iP^{(1)}_{k-2}).$$

Consequently, by applying (2.1), we have the relation for $i = 0, 1, ..., k-1$

$$A_kc(x^{i+4}P_{k-4}) + B_kc(x^{i+3}P_{k-4}) + C_kc(x^{i+2}P_{k-4}) + D_kc(x^{i+1}P_{k-4}) + E_kc(x^iP_{k-4})+$$
$$F_kc^{(1)}(x^{i+1}P^{(1)}_{k-2}) + G_kc^{(1)}(x^iP^{(1)}_{k-2}) + H_kc(x^iP^{(1)}_{k-2}) = 0. \tag{2.19}$$

For $i = 0$, equation (2.19) gives

$$H_kc(x^0P^{(1)}_{k-2}) = 0 \;\Rightarrow\; c(P^{(1)}_{k-2}) \neq 0,\; H_k = 0.$$

Hence from (2.18), we have $E_k = 1$. For $i = 0, 1, 2, ..., k-9$, the relation (2.19) is always true. Therefore for $i = k-8$, equation (2.19) gives

$$A_kc(x^{k-4}P_{k-4}) = 0 \;\Rightarrow\; c(x^{k-4}P_{k-4}) \neq 0,\; A_k = 0.$$

For $i = k-7$, equation (2.19) gives

$$B_kc(x^{k-4}P_{k-4}) = 0 \;\Rightarrow\; c(x^{k-4}P_{k-4}) \neq 0,\; B_k = 0.$$

For $i = k-6$, equation (2.19) gives

$$C_kc(x^{k-4}P_{k-4}) = 0 \;\Rightarrow\; c(x^{k-4}P_{k-4}) \neq 0,\; C_k = 0.$$

For $i = k-5$, equation (2.19) gives

$$D_kc(x^{k-4}P_{k-4}) = 0 \;\Rightarrow\; c(x^{k-4}P_{k-4}) \neq 0,\; D_k = 0.$$

For $i = k-4$ and $E_k = 1$, equation (2.19) gives

$$E_kc(x^{k-4}P_{k-4}) = 0 \;\Rightarrow\; c(x^{k-4}P_{k-4}) = 0.$$

This is impossible from condition (2.1). Therefore the formula $A_{21}$ does not exist and consequently algorithm $A_{21}$ does not exist too.

### 2.2.3 $A_{22}$ for $U_i(x) = x^i$

Consider the following recurrence relationship for $k \geq 4$,

$$P_k(x) = A_k\Big\{(x^3 + B_k x^2 + C_k x + D_k)P_{k-3} + (E_k x^4 + F_k x^3 + G_k x^2 + H_k x + I_k)P^{(1)}_{k-4}\Big\}, \tag{2.20}$$

where $P_k(x)$, $P_{k-3}(x)$ and $P^{(1)}_{k-4}(x)$ are polynomials of degree $k$, $k-3$ and $k-4$ respectively. The constant coefficients $A_k$, $B_k$, $C_k$, $D_k$, $E_k$,$F_k$, $G_k$, $H_k$ and $I_k$ are determined by $P_k(0) = 1$ and imposing the orthogonality condition (2.1). Since $P_k(0) = 1$, $\forall\ k$, then for $x = 0$, equation (2.20) becomes

$$A_k\{D_k + I_k P^{(1)}_{k-4}(0)\} = 1. \tag{2.21}$$

After multiplying by $x^i$ and applying linear functional $c$ on both sides it becomes

$$c(x^i P_k) = A_k\Big\{c(x^{i+3}P_{k-3}) + B_k c(x^{i+2}P_{k-3}) + C_k c(x^{i+1}P_{k-3}) + D_k c(x^i P_{k-3}) + E_k c(x^{i+4}P^{(1)}_{k-4}) + F_k c(x^{i+3}P^{(1)}_{k-4}) + G_k c(x^{i+2}P^{(1)}_{k-4}) + H_k c(x^{i+1}P^{(1)}_{k-4}) + I_k c(x^i P^{(1)}_{k-4})\Big\}. \tag{2.22}$$

Consequently, by applying (2.1), we have the relation for $i = 0, 1, ..., k-1$

$$c(x^{i+3}P_{k-3}) + B_k c(x^{i+2}P_{k-3}) + C_k c(x^{i+1}P_{k-3}) + D_k c(x^i P_{k-3}) + E_k c^{(1)}(x^{i+3}P^{(1)}_{k-4}) + F_k c^{(1)}(x^{i+2}P^{(1)}_{k-4}) + G_k c^{(1)}(x^{i+1}P^{(1)}_{k-4}) + H_k c^{(1)}(x^i P^{(1)}_{k-4}) + I_k c(x^i P^{(1)}_{k-4}) = 0. \tag{2.23}$$

For $i = 0$, equation (2.23) gives

$$I_k c(x^0 P^{(1)}_{k-4}) = 0, \quad \Rightarrow \quad c(P^{(1)}_{k-4}) \neq 0, \quad I_k = 0.$$

Hence from (2.21), we have

$$A_k = \frac{1}{D_k}. \tag{2.24}$$

The orthogonality condition is always true for $i = 0, 1, 2, ......, k-8$. Therefore for $i = k-7$, equation (2.23) gives

$$E_k c^{(1)}(x^{k-4}P^{(1)}_{k-4}) = 0, \quad \Rightarrow \quad c^{(1)}(x^{k-4}P^{(1)}_{k-4}) \neq 0, \quad E_k = 0.$$

For $i = k-6$, equation (2.23) gives

$$F_k = -\frac{c(x^{k-3}P_{k-3})}{c(x^{k-3}P^{(1)}_{k-4})}. \tag{2.25}$$

For $i = k-5$, equation (2.23) gives

$$B_k c(x^{k-3}P_{k-3}) + G_k c^{(1)}(x^{k-4}P^{(1)}_{k-4}) = -c(x^{k-2}P_{k-3}) - F_k c^{(1)}(x^{k-3}P^{(1)}_{k-4}). \tag{2.26}$$

For $i = k-4$, equation (2.23) gives

$$B_k c(x^{k-2}P_{k-3}) + C_k c(x^{k-3}P_{k-3}) + G_k c^{(1)}(x^{k-3}P^{(1)}_{k-4}) + H_k c^{(1)}(x^{k-4}P^{(1)}_{k-4}) = -c(x^{k-1}P_{k-3}) - F_k c^{(1)}(x^{k-2}P^{(1)}_{k-4}). \tag{2.27}$$

For $i = k-3$, and equation (2.23) gives

$$\begin{aligned} B_k c(x^{k-1}P_{k-3}) + C_k c(x^{k-2}P_{k-3}) + D_k c(x^{k-3}P_{k-3}) + G_k c^{(1)}(x^{k-2}P^{(1)}_{k-4} + H_k c^{(1)}(x^{k-3}P^{(1)}_{k-4}) \\ = -c(x^{k}P_{k-3}) - F_k c^{(1)}(x^{k-1}P^{(1)}_{k-4}). \end{aligned} \tag{2.28}$$

For $i = k-2$ and equation (2.23) gives

$$\begin{aligned} B_k c(x^{k}P_{k-3}) + C_k c(x^{k-1}P_{k-3}) + D_k c(x^{k-2}P_{k-3}) + G_k c^{(1)}(x^{k-1}P^{(1)}_{k-4}) + H_k c^{(1)}(x^{k-2}P^{(1)}_{k-4}) \\ = -c(x^{k+1}P_{k-3}) - F_k c^{(1)}(x^{k}P^{(1)}_{k-4}). \end{aligned} \tag{2.29}$$

For $i = k-1$ and equation (2.23) gives

$$\begin{aligned} B_k c(x^{k+1}P_{k-3}) + C_k c(x^{k}P_{k-3}) + D_k c(x^{k-1}P_{k-3}) + G_k c^{(1)}(x^{k}P^{(1)}_{k-4}) + H_k c^{(1)}(x^{k-1}P^{(1)}_{k-4}) \\ = -c(x^{k+2}P_{k-3}) - F_k c^{(1)}(x^{k+1}P^{(1)}_{k-4}). \end{aligned} \tag{2.30}$$

Equations (2.26), (2.27), (2.28), (2.29) and (2.30) can be written as

$$\begin{cases} a_{11}B_k + a_{14}G_k = b_1, \\ a_{21}B_k + a_{22}C_k + a_{24}G_k + a_{25}H_k = b_2, \\ a_{31}B_k + a_{32}C_k + a_{33}D_k + a_{34}G_k + a_{35}H_k = b_3, \\ a_{41}B_k + a_{42}C_k + a_{43}D_k + a_{44}G_k + a_{45}H_k = b_4, \\ a_{51}B_k + a_{52}C_k + a_{53}D_k + a_{54}G_k + a_{55}H_k = b_5. \end{cases} \tag{2.31}$$

Where $a_{11}$, $a_{14}$, $a_{21}$, $a_{22}$, $a_{24}$, $a_{25}$, $a_{31}$, $a_{32}$, $a_{33}$, $a_{34}$, $a_{35}$, $a_{41}$, $a_{42}$, $a_{43}$, $a_{44}$, $a_{45}$, $a_{51}$, $a_{52}$, $a_{53}$, $a_{54}$, and $a_{55}$ are the coefficients of $B_k$, $C_k$, $D_k$, $G_k$ and $H_k$ respectively. Suppose $b_1$, $b_2$, $b_3$, $b_4$, and $b_5$ are the corresponding right hand side terms of these equations. If $\Delta_k$ represents the determinant of the coefficients matrix of (2.31). From (2.14), if $\Delta_k \neq 0$, then

$$\begin{cases} B_k, C_k \text{ as in (2.15)} \\ G_k = \frac{b_1 - a_{11}B_k}{a_{14}}, \\ H_k = \frac{b_2 - a_{21}B_k - a_{22}C_k - a_{24}G_k}{a_{25}}, \\ D_k = \frac{b_3 - a_{31}B_k - a_{32}C_k - a_{34}G_k - a_{35}H_k}{a_{33}}. \end{cases} \tag{2.32}$$

Since $E_k = I_k = 0$, relation $A_{22}$ becomes

$$P_k(x) = A_k\Big\{(x^3 + B_kx^2 + C_kx + D_k)P_{k-3}(x) + (F_kx^3 + G_kx^2 + H_kx)P^{(1)}_{k-4}(x)\Big\}. \tag{2.33}$$

Therefore, $A_{22}$ leads to a Lanczos-type algorithm.

### 2.2.4 $A_{23}$ for $U_i(x) = x^i$

Consider the following recurrence relationship for $k \geq 4$,

$$P_k(x) = A_k\Big\{(x^3 + B_kx^2 + C_kx + D_k)P^{(1)}_{k-3} + (E_kx^4 + F_kx^3 + G_kx^2 + H_kx + I_k)P^{(1)}_{k-4}\Big\}, \tag{2.34}$$

where $P_k(x)$, $P^{(1)}_{k-3}(x)$ and $P^{(1)}_{k-4}(x)$ are polynomials of degree $k$, $k-3$ and $k-4$ respectively. The constant coefficients $A_k$, $B_k$, $C_k$, $D_k$, $E_k$,$F_k$, $G_k$, $H_k$ and $I_k$ are determined by $P_k(0) = 1$ and imposing the orthogonality condition (2.1). Since $P_k(0) = 1$, $\forall\, k$, then for $x = 0$, equation (2.34) becomes

$$A_k\{D_kP^{(1)}_{k-3} + I_kP^{(1)}_{k-4}(0)\} = 1. \tag{2.35}$$

After multiplying by $x^i$ and applying linear functional $c$ on both sides it becomes

$$\begin{aligned} c(x^iP_k) = \; & A_k\Big\{c(x^{i+3}P^{(1)}_{k-3}) + B_kc(x^{i+2}P^{(1)}_{k-3}) + C_kc(x^{i+1}P^{(1)}_{k-3}) + D_kc(x^iP^{(1)}_{k-3}) + E_kc(x^{i+4}P^{(1)}_{k-4}) \\ & + F_kc(x^{i+3}P^{(1)}_{k-4}) + G_kc(x^{i+2}P^{(1)}_{k-4}) + H_kc(x^{i+1}P^{(1)}_{k-4}) + I_kc(x^iP^{(1)}_{k-4})\Big\}. \end{aligned} \tag{2.36}$$

Consequently, by applying (2.1), we have the relation for $i = 0, 1, ..., k-1$

$$c^{(1)}(x^{i+2}P^{(1)}_{k-3}) + B_k c^{(1)}(x^{i+1}P^{(1)}_{k-3}) + C_k c^{(1)}(x^{i}P^{(1)}_{k-3}) + D_k c(x^{i}P^{(1)}_{k-3}) + E_k c^{(1)}(x^{i+3}P^{(1)}_{k-4})$$
$$+F_k c^{(1)}(x^{i+2}P^{(1)}_{k-4}) + G_k c^{(1)}(x^{i+1}P^{(1)}_{k-4}) + H_k c^{(1)}(x^{i}P^{(1)}_{k-4}) + I_k c(x^{i}P^{(1)}_{k-4}) = 0. \tag{2.37}$$

For $i = 0$, equation (2.37) gives

$$D_k c(P^{(1)}_{k-3}) + I_k c(P^{(1)}_{k-4}) = 0. \tag{2.38}$$

The orthogonality condition is always true for $i = 0, 1, 2, ......, k-8$. Therefore for $i = k-7$, equation (2.37) gives

$$E_k c^{(1)}(x^{k-4}P^{(1)}_{k-4}) = 0 \quad \Rightarrow \quad c^{(1)}(x^{k-4}P^{(1)}_{k-4}) \neq 0, \quad E_k = 0.$$

For $i = k-6$, equation (2.37) gives

$$F_k c^{(1)}(x^{k-4}P^{(1)}_{k-4}) = 0, \quad \Rightarrow \quad c^{(1)}(x^{k-4}P^{(1)}_{k-4}) \neq 0, \quad F_k = 0.$$

For $i = k-5$, equation (2.37) gives

$$c^{(1)}(x^{k-3}P^{(1)}_{k-3}) + G_k c^{(1)}(x^{k-4}P^{(1)}_{k-4}) = 0,$$
$$G_k = \frac{-c(x^{k-2}P^{(1)}_{k-3})}{c(x^{k-3}P^{(1)}_{k-4})}. \tag{2.39}$$

For $i = k-4$, equation (2.37) gives

$$B_k c^{(1)}(x^{k-3}P^{(1)}_{k-3}) + H_k c^{(1)}(x^{k-4}P^{(1)}_{k-4}) + I_k c(x^{k-4}P^{(1)}_{k-4}) = -c(x^{k-1}P_{k-3}) - G_k c^{(1)}(x^{k-3}P^{(1)}_{k-4}). \tag{2.40}$$

For $i = k-3$, and equation (2.37) gives

$$B_k c^{(1)}(x^{k-2}P^{(1)}_{k-3}) + C_k c^{(1)}(x^{k-3}P^{(1)}_{k-3}) + D_k c(x^{k-3}P^{(1)}_{k-3}) + H_k c^{(1)}(x^{k-3}P^{(1)}_{k-4} + I_k c(x^{k-3}P^{(1)}_{k-4})$$
$$= -c^{(1)}(x^{k-1}P^{(1)}_{k-3}) - G_k c^{(1)}(x^{k-2}P^{(1)}_{k-4}). \tag{2.41}$$

For $i = k-2$, and equation (2.37) gives

$$B_k c^{(1)}(x^{k-1}P^{(1)}_{k-3}) + C_k c^{(1)}(x^{k-2}P^{(1)}_{k-3}) + D_k c(x^{k-2}P^{(1)}_{k-3}) + H_k c^{(1)}(x^{k-2}P^{(1)}_{k-4} + I_k c(x^{k-2}P^{(1)}_{k-4})$$
$$= -c^{(1)}(x^{k}P^{(1)}_{k-3}) - G_k c^{(1)}(x^{k-1}P^{(1)}_{k-4}). \tag{2.42}$$

For $i = k-1$, and equation (2.37) gives

$$B_k c^{(1)}(x^k P_{k-3}^{(1)}) + C_k c^{(1)}(x^{k-1} P_{k-3}^{(1)}) + D_k c(x^{k-1} P_{k-3}^{(1)}) + H_k c^{(1)}(x^{k-1} P_{k-4}^{(1)} + I_k c(x^{k-1} P_{k-4}^{(1)}) = -c^{(1)}(x^{k+1} P_{k-3}^{(1)}) - G_k c^{(1)}(x^k P_{k-4}^{(1)}). \tag{2.43}$$

The values of constant coefficients $A_k$, $B_k$, $C_k$, $D_k$, $G_k$, $H_k$ and $I_k$ can be obtained by solving the equations (2.38), (2.40), (2.41), (2.42) and (2.43). Since $E_k = F_k = 0$, relation $A_{23}$ becomes

$$P_k(x) = A_k\Big\{(x^3 + B_k x^2 + C_k x + D_k)P_{k-3}^{(1)} + (G_k x^2 + H_k x + I_k)P_{k-4}^{(1)}\Big\}. \tag{2.44}$$

Since $\mathbf{r}_k = P_k(A)\mathbf{r}_0$, the equation (2.44), after replacing $x$ by $A$, becomes

$$\mathbf{r}_k = A_k\Big\{(A^3 + B_k A^2 + C_k A + D_k)\mathbf{z}_{k-3} + (G_k A^2 + H_k A + I_k)\mathbf{z}_{k-4}\Big\}. \tag{2.45}$$

Using $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, we get

$$A\mathbf{x}_k = \mathbf{b} - A_k\Big\{(A^3 + B_k A^2 + C_k A + D_k)\mathbf{z}_{k-3} + (G_k A^2 + H_k A + I_k)\mathbf{z}_{k-4}\Big\}. \tag{2.46}$$

It is clear from the above equation (2.46) that we cannot find $x_k$ from $r_k$ without inverting $A$. So, a Lanczos algorithm based on $A_{23}$ cannot be implemented.

### 2.2.5 $A_{24}$ for $U_i(x) = x^i$

Consider the following recurrence relationship for $k \geq 4$

$$P_k(x) = (A_k x^4 + B_k x^3 + C_k x^2 + D_k x + E_k)P_{k-4}^{(1)} + (F_k x^2 + G_k x + H_k)P_{k-2}^{(1)}, \tag{2.47}$$

where $P_k$, $P_{k-2}^{(1)}$ and $P_{k-4}^{(1)}$ are polynomials of degree $k$, $k-1$ and $k-4$ respectively. The constant coefficients $A_k$, $B_k$, $C_k$, $D_k$, $E_k$ $F_k$ $G_k$ and $H_k$ are determined by $P_k(0) = 1$ and imposing the orthogonality condition (2.1). Since $P_k(0) = 1$, $\forall\ k$, then for $x = 0$, equation (2.47) becomes

$$E_k P_{k-4}^{(1)}(0) + H_k P_{k-2}^{(1)}(0) = 1. \tag{2.48}$$

After multiplying equation (2.47) by $x^i$ and applying linear functional $c$ on both sides it becomes

$$c(x^iP_k) = A_kc(x^{i+4}P^{(1)}_{k-4}) + B_kc(x^{i+3}P^{(1)}_{k-4}) + C_kc(x^{i+2}P^{(1)}_{k-4}) + D_kc(x^{i+1}P^{(1)}_{k-4}) + E_kc(x^iP^{(1)}_{k-4}) +$$
$$F_kc(x^{i+2}P^{(1)}_{k-2}) + G_kc(x^{i+1}P^{(1)}_{k-2}) + H_kc(x^iP^{(1)}_{k-2}).$$

Consequently, by applying (2.1), we have the relation for $i = 0, 1, ..., k-1$

$$A_kc^{(1)}(x^{i+3}P^{(1)}_{k-4}) + B_kc^{(1)}(x^{i+2}P^{(1)}_{k-4}) + C_kc^{(1)}(x^{i+1}P^{(1)}_{k-4}) + D_kc^{(1)}(x^iP^{(1)}_{k-4}) + E_kc(x^iP^{(1)}_{k-4}) +$$
$$F_kc^{(1)}(x^{i+1}P^{(1)}_{k-2}) + G_kc^{(1)}(x^iP^{(1)}_{k-2}) + H_kc(x^iP^{(1)}_{k-2}) = 0. \tag{2.49}$$

For $i = 0$, equation (2.49) gives

$$E_kc(P^{(1)}_{k-4}) + H_kc(P^{(1)}_{k-2}) = 0. \tag{2.50}$$

For $i = 0, 1, 2, ..., k-8$, the relation (2.49) is always true. Therefore for $i = k-7$, equation (2.49) gives

$$A_kc^{(1)}(x^{k-4}P^{(1)}_{k-4}) = 0 \;\Rightarrow\; c^{(1)}(x^{k-4}P^{(1)}_{k-4}) \neq 0,\; A_k = 0.$$

For $i = k-6$, equation (2.49) gives

$$B_kc^{(1)}(x^{k-4}P^{(1)}_{k-4}) = 0 \;\Rightarrow\; c^{(1)}(x^{k-4}P^{(1)}_{k-4}) \neq 0,\; B_k = 0.$$

For $i = k-5$, equation (2.49) gives

$$C_kc^{(1)}(x^{k-4}P^{(1)}_{k-4}) = 0 \;\Rightarrow\; c^{(1)}(x^{k-4}P^{(1)}_{k-4}) \neq 0,\; C_k = 0.$$

For $i = k-4$, equation (2.49) gives

$$D_kc^{(1)}(x^{k-4}P^{(1)}_{k-4}) + E_kc(x^{k-4}P^{(1)}_{k-4} = 0. \tag{2.51}$$

For $i = k-3$, equation (2.49) gives

$$D_kc^{(1)}(x^{k-3}P^{(1)}_{k-4}) + E_kc(x^{k-3}P^{(1)}_{k-4}) + F_kc^{(1)}(x^{k-2}P^{(1)}_{k-2}) = 0. \tag{2.52}$$

For $i = k-2$, equation (2.49) gives

$$D_k c^{(1)}(x^{k-2}P^{(1)}_{k-4}) + E_k c(x^{k-2}P^{(1)}_{k-4}) + F_k c^{(1)}(x^{k-1}P^{(1)}_{k-2}) + G_k c^{(1)}(x^{k-2}P^{(1)}_{k-2}) + H_k c(x^{k-2}P^{(1)}_{k-2}) = 0. \quad (2.53)$$

For $i = k-1$, equation (2.49) gives

$$D_k c^{(1)}(x^{k-1}P^{(1)}_{k-4}) + E_k c(x^{k-1}P^{(1)}_{k-4}) + F_k c^{(1)}(x^{k}P^{(1)}_{k-2}) + G_k c^{(1)}(x^{k-1}P^{(1)}_{k-2}) + H_k c(x^{k-1}P^{(1)}_{k-2}) = 0. \quad (2.54)$$

Hence, we have six equations (2.48), (2.50), (2.51), (2.52), (2.53) and (2.54) to find five unknown constants $D_k$, $E_k$, $F_k$, $G_k$ and $H_k$, showing that the system is overdetermined. The recurrence relation $A_{24}$ therefore cannot be used to implement a Lanczos-type algorithm. Since $A_k = B_k = C_k = 0$, relation $A_{24}$ becomes

$$P_k(x) = (D_k x + E_k)P^{(1)}_{k-4} + (F_k x^2 + G_k x + H_k)P^{(1)}_{k-2}. \quad (2.55)$$

One more reason which explains, why we cannot use the relation $A_{24}$ for the implementation of a Lanczos-type algorithm, even if the above relationship is perfectly valid and exists, is as follows. Multiplying both sides of equation (2.55) by $\mathbf{r}_0$, after replacing $x$ by $A$ and simplifying by using $\mathbf{r}_k = P_k(A)\mathbf{r}_0$ and $\mathbf{z}_k = P^{(1)}_k(A)\mathbf{r}_0$, we have

$$\mathbf{r}_k = (A_k x^4 + B_k x^3 + C_k x^2 + D_k x + E_k)\mathbf{z}_{k-4} + (F_k x^2 + G_k x + H_k)\mathbf{z}_{k-2}. \quad (2.56)$$

Using $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, we get

$$\mathbf{x}_k = A^{-1}\mathbf{b} - A^{-1}(A_k x^4 + B_k x^3 + C_k x^2 + D_k x + E_k)\mathbf{z}_{k-4} + (F_k x^2 + G_k x + H_k)\mathbf{z}_{k-2}. \quad (2.57)$$

It is clear from equation (2.57) that we cannot find $\mathbf{x}_k$ from $\mathbf{r}_k$ without inverting $A$. So, this relation is not desirable for implementing a Lanczos-type algorithm as it involves a matrix inversion.

### 2.2.6 $A_{25}$ for $U_i(x) = x^i$

Consider the following recurrence relationship for $k \geq 3$

$$P_k(x) = (A_k x^3 + B_k x^2 + C_k x + D_k)P_{k-3} + (E_k x^3 + F_k x^2 + G_k x + H_k)P^{(1)}_{k-3}, \tag{2.58}$$

where $P_k$, $P^{(1)}_{k-3}$ and $P_{k-3}$ are polynomials of degree $k$, $k-3$ and $k-3$ respectively. The constant coefficients $A_k$, $B_k$, $C_k$, $D_k$, $E_k$, $F_k$ and $G_k$ are determined by $P_k(0) = 1$ and imposing the orthogonality condition (2.1). Since $P_k(0) = 1$, $\forall\ k$, then for $x = 0$, equation (2.58) becomes

$$D_k + H_k P^{(1)}_{k-3}(0) = 1. \tag{2.59}$$

After multiplying equation (2.58) by $x^i$ and applying linear functional $c$ on both sides it becomes

$$c(x^i P_k) = A_k c(x^{i+3} P_{k-3}) + B_k c(x^{i+2} P_{k-3}) + C_k c(x^{i+1} P_{k-3}) + D_k c(x^i P_{k-3}) + E_k c(x^{i+3} P^{(1)}_{k-3}) +$$
$$F_k c(x^{i+2} P^{(1)}_{k-3}) + G_k c(x^{i+1} P^{(1)}_{k-3}) + H_k c(x^i P^{(1)}_{k-3}).$$

Consequently, by applying (2.1), we have the relation for $i = 0, 1, ..., k-1$

$$A_k c(x^{i+3} P_{k-3}) + B_k c(x^{i+2} P_{k-3}) + C_k c(x^{i+1} P_{k-3}) + D_k c(x^i P_{k-3}) + E_k c^{(1)}(x^{i+2} P_{k-3}) +$$
$$F_k c^{(1)}(x^{i+1} P^{(1)}_{k-3}) + G_k c^{(1)}(x^i P^{(1)}_{k-3}) + H_k c(x^i P^{(1)}_{k-3}) = 0. \tag{2.60}$$

For $i = 0$, equation (2.60) gives

$$H_k c(x^0 P^{(1)}_{k-3}) = 0 \quad \Rightarrow \quad c(P^{(1)}_{k-3}) \neq 0, \quad H_k = 0.$$

Hence from (2.59), we have $D_k = 1$. For $i = 0, 1, 2, ..., k-7$, the relation (2.60) is always true. Therefore for $i = k-6$, equation (2.60) gives

$$A_k c(x^{k-3} P_{k-3}) = 0 \quad \Rightarrow \quad c(x^{k-3} P_{k-3}) \neq 0, \quad A_k = 0. \tag{2.61}$$

For $i = k-5$, equation (2.60) gives

$$B_k c(x^{k-3} P_{k-3}) + E_k c^{(1)}(x^{k-3} P^{(1)}_{k-3}) = 0. \tag{2.62}$$

For $i = k - 4$, equation (2.60) gives

$$B_k c(x^{k-2}P_{k-3}) + C_k c(x^{k-3}P_{k-3}) + E_k c^{(1)}(x^{k-2}P^{(1)}_{k-3}) + F_k c^{(1)}(x^{k-3}P^{(1)}_{k-3}) = 0. \quad (2.63)$$

For $i = k - 3$, equation (2.60) gives

$$B_k c(x^{k-1}P_{k-3}) + C_k c(x^{k-2}P_{k-3}) + E_k c^{(1)}(x^{k-1}P^{(1)}_{k-3}) + F_k c^{(1)}(x^{k-2}P^{(1)}_{k-3}) + G_k c^{(1)}(x^{k-3}P^{(1)}_{k-3}) = -c(x^{k-3}P_{k-3}). \quad (2.64)$$

For $i = k - 2$, equation (2.60) gives

$$B_k c(x^{k}P_{k-3}) + C_k c(x^{k-1}P_{k-3}) + E_k c^{(1)}(x^{k}P^{(1)}_{k-3}) + F_k c^{(1)}(x^{k-1}P^{(1)}_{k-3}) + G_k C^{(1)}(x^{k-2}P^{(1)}_{k-3}) = -c(x^{k-2}P_{k-3}). \quad (2.65)$$

For $i = k - 1$, equation (2.60) gives

$$B_k c(x^{k+1}P_{k-3}) + C_k c(x^{k}P_{k-3}) + E_k c^{(1)}(x^{k+1}P^{(1)}_{k-3}) + F_k c^{(1)}(x^{k}P^{(1)}_{k-3}) + G_k C^{(1)}(x^{k-1}P^{(1)}_{k-3}) = -c(x^{k-1}P_{k-3}). \quad (2.66)$$

Equations (2.62), (2.63), (2.64), (2.65) and (2.66) can be written as

$$\begin{cases} a_{11}B_k + a_{13}E_k = 0, \\ a_{21}B_k + a_{22}C_k + a_{23}E_k + a_{24}F_k = 0, \\ a_{31}B_k + a_{32}C_k + a_{33}E_k + a_{34}F_k + G_k a_{35} = b_3, \\ a_{41}B_k + a_{42}C_k + a_{43}E_k + a_{44}F_k + G_k a_{45} = b_4, \\ a_{51}B_k + a_{52}C_k + a_{53}E_k + a_{54}F_k + G_k a_{55} = b_5. \end{cases} \quad (2.67)$$

Where $a_{11}$, $a_{13}$, $a_{21}$, $a_{22}$, $a_{23}$, $a_{24}$, $a_{31}$, $a_{32}$, $a_{33}$, $a_{34}$, $a_{35}$, $a_{41}$, $a_{42}$, $a_{43}$, $a_{44}$, $a_{45}$, $a_{51}$, $a_{52}$, $a_{53}$, $a_{54}$, and $a_{55}$ are the coefficients of $B_k$, $C_k$, $E_k$, $F_k$ and $G_k$ respectively. Suppose $b_3$, $b_4$, and $b_5$ are the corresponding right hand side terms of these equations. If $\Delta_k$ represents the determinant of the coefficients matrix of (2.67) then we have,

$$\Delta_k = det(Q), \quad (2.68)$$

where $Q = matrix([q_1, q_2, q_3, q_4, q_5])$,

$q_1 = [a_{11}, 0, a_{13}, 0, 0]$, $q_2 = [a_{21}, a_{22}, a_{23}, a_{24}, 0]$, $q_3 = [a_{31}, a_{32}, a_{33}, a_{34}, a_{35}]$,

$q_4 = [a_{41}, a_{42}, a_{43}, a_{44}, a_{45}]$, $q_5 = [a_{51}, a_{52}, a_{53}, a_{54}, a_{55}]$.

If $\Delta_k \neq 0$, then

$$\begin{cases} B_k = \frac{det(S)}{\Delta_k}, \quad \text{where } S = matrix([s_1, s_2, s_3, s_4, s_5]), \\ s_1 = [0, 0, a_{13}, 0, 0],\ s_2 = [0, a_{22}, a_{23}, a_{24}, 0],\ s_3 = [b_3, a_{32}, a_{33}, a_{34}, a_{35}], \\ s_4 = [b_4, a_{42}, a_{43}, a_{44}, a_{45}],\ s_5 = [b_5, a_{52}, a_{53}, a_{54}, a_{55}], \\ C_k = \frac{det(T)}{\Delta_k}, \quad \text{where } T = matrix([t_1, t_2, t_3, t_4, t_5]), \\ t_1 = [a_{11}, 0, a_{13}, 0, 0],\ t_2 = [a_{21}, 0, a_{23}, a_{24}, 0],\ t_3 = [a_{31}, b_3, a_{33}, a_{34}, a_{35}], \\ t_4 = [a_{41}, b_4, a_{43}, a_{44}, a_{45}],\ t_5 = [a_{51}, b_5, a_{53}, a_{54}, a_{55}], \\ E_k = -\frac{a_{11}B_k}{a_{13}}, \\ F_k = -\frac{a_{21}B_k + a_{22}C_k + a_{23}E_k}{a_{24}}, \\ G_k = \frac{b_3 - a_{31}B_k - a_{32}C_k - a_{33}E_k - a_{34}F_k}{a_{35}}. \end{cases} \tag{2.69}$$

Since $A_k = H_k = 0$ and $D_k = 1$, relation $A_{25}$ becomes

$$P_k(x) = (B_k x^2 + C_k x + I)P_{k-3}(x) + (E_k x^3 + F_k x^2 + G_k x)P^{(1)}_{k-3}(x). \tag{2.70}$$

Therefore $A_{25}$ can lead to a Lanczos-type algorithm.

### 2.2.7 $A_{26}$ for $U_i(x) = x^i$

Let $P_k(x)$, $P^{(1)}_{k-2}(x)$ and $P_{k-3}(x)$ be the orthogonal polynomials of degree $k$, $k-2$ and $k-3$ respectively.

Consider the following recurrence relationship for $k \geq 3$

$$P_k(x) = (A_k x^3 + B_k x^2 + C_k x + D_k)P_{k-3} + (E_k x^2 + F_k x + G_k)P^{(1)}_{k-2}. \tag{2.71}$$

The constant coefficients $A_k$, $B_k$, $C_k$, $D_k$, $E_k$ $F_k$ and $G_k$ are determined by $P_k(0) = 1$ and imposing the orthogonality condition (2.1). Since $P_k(0) = 1$, $\forall\ k$, then for $x = 0$, equation (2.71) becomes

$$D_k + G_k P_{k-2}^{(1)}(0) = 1. \tag{2.72}$$

After multiplying equation (2.71) by $x^i$ and applying linear functional $c$ on both sides it becomes

$$c(x^i P_k) = A_k c(x^{i+3} P_{k-3}) + B_k c(x^{i+2} P_{k-3}) + C_k c(x^{i+1} P_{k-3}) + D_k c(x^i P_{k-3}) + E_k c(x^{i+2} P_{k-2}^{(1)})$$
$$+ F_k c(x^{i+1} P_{k-2}^{(1)}) + G_k c(x^i P_{k-2}^{(1)}).$$

Consequently, by applying (2.1), we have the relation for $i = 0, 1, ..., k-1$

$$A_k c(x^{i+3} P_{k-3}) + B_k c(x^{i+2} P_{k-3}) + C_k c(x^{i+1} P_{k-3}) + D_k c(x^i P_{k-3}) + E_k c^{(1)}(x^{i+1} P_{k-2}^{(1)})$$
$$+ F_k c^{(1)}(x^i P_{k-2}^{(1)}) + G_k c(x^i P_{k-2}^{(1)}) = 0. \tag{2.73}$$

For $i = 0$, equation (2.73) gives

$$G_k c(x^0 P_{k-2}^{(1)}) = 0 \quad \Rightarrow \quad c(P_{k-2}^{(1)}) \neq 0, \quad G_k = 0.$$

Hence from (2.72), we have $D_k = 1$. For $i = 0, 1, 2, ..., k-7$, the relation (2.73) is always true. Therefore for $i = k-6$, equation (2.73) gives

$$A_k c(x^{k-3} P_{k-3}) = 0 \ \Rightarrow \ c(x^{k-3} P_{m-3}) \neq 0, \ A_k = 0.$$

For $i = k-5$, equation (2.73) gives

$$B_k c(x^{k-3} P_{k-3}) = 0 \ \Rightarrow \ c(x^{k-3} P_{k-3}) \neq 0, \ B_k = 0.$$

For $i = k-4$, equation (2.73) gives

$$C_k c(x^{k-3} P_{k-3}) = 0, \ \Rightarrow \ c(x^{k-3} P_{k-3}) \neq 0, \ C_k = 0.$$

For $i = k-3$, equation (2.73) gives

$$D_k c(x^{k-3} P_{k-3}) + E_k c^{(1)}(x^{k-2} P_{k-2}^{(1)}) = 0.$$

Since $D_k = 1$, then

$$E_k = \frac{c(x^{k-3} P_{k-3})}{c^{(1)}(x^{k-2} P_{k-2}^{(1)})}.$$

For $i = k-2$, equation (2.73) gives

$$F_k = \frac{-c(x^{k-2} P_{k-3}) - E_k c^{(1)}(x^{k-1} P_{k-2}^{(1)})}{c^{(1)}(x^{k-2} P_{k-2}^{(1)})}.$$

For $i = k-1$, equation (2.73) gives

$$F_k = \frac{-c(x^{k-1} P_{k-3}) - E_k c^{(1)}(x^{k} P_{k-2}^{(1)})}{c^{(1)}(x^{k-1} P_{k-2}^{(1)})}.$$

So, due to multiple values for the constant coefficient $F_k$ involved. Therefore, this formula $A_{26}$ is not suitable for the implementation of a Lanczos-type algorithm.

### 2.2.8 $A_{27}$ for $U_i(x) = x^i$

Consider the following recurrence relationship for $k \geq 3$,

$$P_k(x) = A_k\Big\{(x^2 + B_k x + C_k) P_{k-2}^{(1)} + (D_k x^3 + E_k x^2 + F_k x + G_k) P_{k-3}^{(1)}\Big\}, \tag{2.74}$$

where $P_k(x)$, $P_{k-2}^{(1)}$ and $P_{k-3}^{(1)}$ are polynomials of degree $k$, $k-2$ and $k-3$ respectively. The constant coefficients $A_k$, $B_k$, $C_k$, $D_k$, $E_k$,$F_k$, and $G_k$ are determined by $P_k(0) = 1$ and imposing the orthogonality condition (2.1). Since $P_k(0) = 1$, $\forall\ k$, then for $x = 0$, equation (2.74) becomes

$$A_k\{C_k P_{k-2}^{(1)}(0) + G_k P_{k-3}^{(1)}(0)\} = 1. \tag{2.75}$$

After multiplying by $x^i$ and applying linear functional $c$ on both sides it becomes

$$c(x^i P_k) = A_k\Big\{c(x^{i+2} P_{k-2}^{(1)}) + B_k c(x^{i+1} P_{k-2}^{(1)}) + C_k c(x^i P_{k-2}^{(1)}) + D_k c(x^{i+3} P_{k-3}^{(1)}) + E_k c(x^{i+2} P_{k-3}^{(1)})$$
$$+ F_k c(x^{i+1} P_{k-3}^{(1)}) + G_k c(x^i P_{k-3}^{(1)})\Big\}. \tag{2.76}$$

Consequently, by applying (2.1), we have the relation for $i = 0, 1, ..., k-1$

$$B_k c^{(1)}(x^i P^{(1)}_{k-2}) + C_k c(x^i P^{(1)}_{k-2}) + D_k c^{(1)}(x^{i+2} P^{(1)}_{k-3}) + E_k c^{(1)}(x^{i+1} P^{(1)}_{k-3}) + F_k c^{(1)}(x^i P^{(1)}_{k-3})$$
$$+G_k c(x^i P^{(1)}_{k-3}) = -c^{(1)}(x^{i+1} P^{(1)}_{k-2}). \tag{2.77}$$

For $i = 0$, equation (2.77) gives

$$C_k c(P^{(1)}_{k-2}) + G_k c(P^{(1)}_{k-3}) = 0. \tag{2.78}$$

The orthogonality condition is always true for $i = 0, 1, 2, ......, k-6$. Therefore for $i = k-5$, equation (2.77) gives

$$D_k c^{(1)}(x^{k-3} P^{(1)}_{k-3}) = 0 \quad \Rightarrow \quad c^{(1)}(x^{k-3} P^{(1)}_{k-3}) \neq 0, \quad D_k = 0.$$

For $i = k-4$, equation (2.77) gives

$$E_k c^{(1)}(x^{k-3} P^{(1)}_{k-3}) = 0, \quad \Rightarrow \quad c^{(1)}(x^{k-3} P^{(1)}_{k-3}) \neq 0, \quad E_k = 0.$$

For $i = k-3$, equation (2.77) gives

$$F_k c^{(1)}(x^{k-3} P_{k-3}) + G_k c(x^{k-3} P^{(1)}_{k-3}) = -c(x^{k-2} P_{k-2}). \tag{2.79}$$

For $i = k-2$, equation (2.77) gives

$$B_k c^{(1)}(x^{k-2} P^{(1)}_{k-2}) + C_k c(x^{k-2} P^{(1)}_{k-2}) + F_k c^{(1)}(x^{k-2} P^{(1)}_{k-3}) + G_k c(x^{k-2} P^{(1)}_{k-3}) = -c^{(1)}(x^{k-1} P^{(1)}_{k-2}). \tag{2.80}$$

For $i = k-1$, and equation (2.77) gives

$$B_k c^{(1)}(x^{k-1} P^{(1)}_{k-2}) + C_k c(x^{k-2} P^{(1)}_{k-2}) + F_k c^{(1)}(x^{k-1} P^{(1)}_{k-3}) + G_k c^{(1)}(x^{k-1} P^{(1)}_{k-3}) = -c^{(1)}(x^k P_{k-2}). \tag{2.81}$$

The values of constant coefficients $A_k$, $B_k$, $C_k$, $F_k$ and $G_k$ can be obtained by solving the equations (2.75), (2.78), (2.79), (2.80) and (2.81). Since $D_k = E_k = 0$, relation $A_{27}$ becomes

$$P_k(x) = A_k \Big\{ (x^2 + B_k x + C_k) P^{(1)}_{k-2} + (F_k x + G_k) P^{(1)}_{k-3} \Big\}. \tag{2.82}$$

Multiplying bothe sides of equation (2.82) by $\mathbf{r}_0$, after replacing $x$ by $A$ and simplifying by using $\mathbf{r}_k = P_k(A)\mathbf{r}_0$, and $\mathbf{z}_k = P_k(A)^{(1)}\mathbf{r}_0$ we have

$$\mathbf{r}_k = A_k \Big\{ (A^2 + B_k A + C_k) \mathbf{z}_{k-2} + (F_k A + G_k I) \mathbf{z}_{k-3} \Big\}. \tag{2.83}$$

Using $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, we get

$$A\mathbf{x}_k = \mathbf{b} - A_k\Big\{(A^2 + B_kA + C_k)\mathbf{z}_{k-2} + (F_kA + G_kI)\mathbf{z}_{k-3}\Big\}. \tag{2.84}$$

It is clear from the above equation (2.84) that we cannot find $\mathbf{x}_k$ from $\mathbf{r}_k$ without inverting $A$. So, the Lanczos algorithm cannot be implemented.

### 2.2.9 $A_{28}$ for $U_i(x) = x^i$

Consider the following recurrence relationship for $k \geq 3$,

$$P_k(x) = A_k\Big\{(x^2 + B_kx + C_k)P_{k-2} + (D_kx^3 + E_kx^2 + F_kx + G_k)P^{(1)}_{k-3}\Big\}, \tag{2.85}$$

where $P_k(x)$, $P_{k-2}(x)$ and $P^{(1)}_{k-3}(x)$ are polynomials of degree $k$, $k-2$ and $k-3$ respectively. The constant coefficients $A_k$, $B_k$, $C_k$, $D_k$, $E_k$,$F_k$ and $G_k$ are determined by $P_k(0) = 1$ and imposing the orthogonality condition (2.1). Since $P_k(0) = 1$, $\forall\ k$, then for $x = 0$, equation (2.85) becomes

$$A_k\{C_k + G_kP^{(1)}_{k-3}\} = 1. \tag{2.86}$$

After multiplying by $x^i$ and applying linear functional $c$ on both sides it becomes

$$c(x^iP_k) = A_k\Big\{c(x^{i+2}P_{k-2}) + B_kc(x^{i+1}P_{k-2}) + C_kc(x^iP_{k-2}) + D_kc(x^{i+3}P^{(1)}_{k-3}) + E_kc(x^{i+2}P^{(1)}_{k-3}) + F_kc(x^{i+1}P^{(1)}_{k-3}) + G_kc(x^iP^{(1)}_{k-3})\Big\}. \tag{2.87}$$

Consequently, by applying (2.1), we have the relation for $i = 0, 1, ..., k-1$

$$c(x^{i+2}P_{k-2}) + B_kc(x^{i+1}P_{k-2}) + C_kc(x^iP_{k-2}) + D_kc(x^{i+3}P^{(1)}_{k-3}) + E_kc(x^{i+2}P^{(1)}_{k-3}) + F_kc(x^{i+1}P^{(1)}_{k-3}) + G_kc(x^iP^{(1)}_{k-3}) = 0. \tag{2.88}$$

For $i = 0$, equation (2.88) becomes

$$G_kc(P^{(1)}_{k-3}) = 0, \quad \text{since} \quad c(P^{(1)}_{k-3}) \neq 0 \quad \Rightarrow \quad G_k = 0.$$

Therefore, from (2.86) we have

$$A_k = \frac{1}{C_k}. \tag{2.89}$$

The orthogonality condition is always true for $i = 0, 1, 2, ......, k-6$. Therefore, for $i = k-5$, equation (2.88) gives, $D_k c^{(1)}(x^{k-3}P^{(1)}_{k-3}) = 0 \quad \Rightarrow \quad c^{(1)}(x^{k-3}P^{(1)}_{k-3}) \neq 0, \quad D_k = 0.$

For $i = k-4$, equation (2.88) gives

$$E_k = -\frac{c(x^{k-2}P_{k-2})}{c^{(1)}(x^{k-3}P^{(1)}_{k-3})}. \tag{2.90}$$

For $i = k-3$, equation (2.88) gives

$$B_k c(x^{k-2}P_{k-2}) + F_k c^{(1)}(x^{k-3}P^{(1)}_{k-3}) = -c(x^{k-1}P_{k-2}) - E_k c^{(1)}(x^{k-2}P^{(1)}_{k-3}). \tag{2.91}$$

For $i = k-2$, equation (2.88) gives

$$B_k c(x^{k-1}P_{k-2}) + C_k c(x^{k-2}P_{k-2}) + F_k c^{(1)}(x^{k-2}P^{(1)}_{k-3}) = -c(x^{k}P_{k-2}) - E_k c^{(1)}(x^{k-1}P^{(1)}_{k-3}). \tag{2.92}$$

For $i = k-1$, and equation (2.88) gives

$$B_k c(x^{k}P_{k-2}) + C_k c(x^{k-1}P_{k-2}) + F_k c^{(1)}(x^{k-1}P^{(1)}_{k-3}) = -c(x^{k+1}P_{k-2}) - E_k c^{(1)}(x^{k}P^{(1)}_{k-3}). \tag{2.93}$$

Equations (2.91), (2.92) and (2.93) can be written as

$$\begin{cases} a_{11}B_k + a_{13}F_k = b_1, \\ a_{21}B_k + a_{22}C_k + a_{23}F_k = b_2, \\ a_{31}B_k + a_{32}C_k + a_{33}F_k = b_3. \end{cases} \tag{2.94}$$

Where $a_{11}$, $a_{13}$, $a_{21}$, $a_{22}$, $a_{23}$, $a_{31}$, $a_{32}$, $a_{33}$, are the coefficients of $B_k$, $C_k$, and $F_k$ respectively. Suppose $b_1$, $b_2$ and $b_3$ are the corresponding right hand side terms of these equations. If $\Delta_k$ represents the determinant of the coefficients matrix of (2.94) then we have,

$$\Delta_k = a_{11}(a_{22}a_{33} - a_{23}a_{32}) + a_{13}(a_{21}a_{32} - a_{31}a_{22}).$$

If $\Delta_k \neq 0$, then

$$\begin{cases} B_k = \frac{1}{\Delta_k}\{b_1(a_{22}a_{33} - a_{23}a_{32}) + a_{13}(b_2a_{32} - b_3a_{22})\}, \\ C_k = \frac{b_2 - a_{21}B_k - F_k a_{23}}{a_{22}}, \\ F_k = \frac{b_1 - a_{11}B_k}{a_{13}}. \end{cases} \tag{2.95}$$

Since $D_k = G_k = 0$, relation $A_{28}$ becomes

$$P_k(x) = A_k\Big\{(x^2 + B_k x + C_k)P_{k-2}(x) + (E_k x^2 + F_k x)P^{(1)}_{k-3}(x)\Big\}. \tag{2.96}$$

Therefore $A_{28}$ can lead to a Lanczos-type algorithm.

## 2.3 Recursive Computation Between the FOPs for $B_i$

Now we consider recurrence relations of the type $B_j$ for the choice $U_i(x) = x^i$. These formulae, when they exist, will be used in combination with formulae $A_i$ to derive Lanczos-type algorithms.

### 2.3.1 $B_{17}$ for $U_i(x) = x^i$

Consider the following recurrence relationship for $k \geq 4$

$$P^{(1)}_k(x) = (A^1_k x^4 + B^1_k x^3 + C^1_k x^2 + D^1_k x + E^1_k)P_{k-4} + (F^1_k x^2 + G^1_k x + H^1_k)P_{k-2}, \tag{2.97}$$

where $P_k(x)$, $P_{k-2}(x)$ and $P_{k-4}(x)$ are polynomials of degree $k$, $k-1$ and $k-4$ respectively. The constant coefficients $A^1_k$, $B^1_k$, $C^1_k$, $D^1_k$, $E^1_k$ $F^1_k$ $G^1_k$ and $H^1_k$ are determined. After multiplying equation (2.97) by $x^i$ and applying linear functional $c^{(1)}$ on both sides it becomes

$$\begin{aligned} c^{(1)}(x^i P_k) = \quad & A^1_k c(x^{i+5}P_{k-4}) + B^1_k c(x^{i+4}P_{k-4}) + C^1_k c(x^{i+3}P_{k-4}) + D^1_k c(x^{i+2}P_{k-4}) + E^1_k c(x^{i+1}P_{k-4}) \\ & + F^1_k c(x^{i+3}P_{k-2}) + G^1_k c(x^{i+2}P_{k-2}) + H^1_k c(x^{i+1}P_{k-2}). \end{aligned}$$

Consequently, by applying (2.2), we have the relation for $i = 0, 1, ..., k-1$.

$$\begin{aligned} A^1_k c(x^{i+5}P_{k-4}) + B^1_k c(x^{i+4}P_{k-4}) + C^1_k c(x^{i+3}P_{k-4}) + D^1_k c(x^{i+2}P_{k-4}) + E^1_k c(x^{i+1}P_{k-4}) + F^1_k c(x^{i+3}P_{k-2}) + \\ G^1_k c(x^{i+2}P_{k-2}) + H^1_k c(x^{i+1}P_{k-2}) = 0. \end{aligned} \tag{2.98}$$

For $i = 0, 1, 2, ..., k-10$, the relation (2.98) is always true. Therefore for $i = k-9$, equation (2.98) gives

$$A^1_k c(x^{k-4}P_{k-4}) = 0 \;\Rightarrow\; c(x^{k-4}P_{k-4}) \neq 0,\; A^1_k = 0.$$

For $i = k-8$, $i = k-7$, and $i = k-6$ equation (2.98) gives $B_k^1 = 0$, $C_k^1 = 0$ and $D_k^1 = 0$.

For $i = k-5$, $i = k-4$, $i = k-3$, $i = k-2$ and $i = k-1$, we get five equations to determine four unknown constant coefficients, $E_k^1$, $F_k^1$, $G_k^1$ and $H_k^1$. This shows that the obtained equations are over-determined, so a Lanczos-type algorithm based on $B_{17}$ cannot be implemented.

### 2.3.2 $B_{18}$ for $U_i(x) = x^i$

Consider the following recurrence relationship for $k \geq 4$,

$$P_k^{(1)} = (A_k^1 x^4 + B_k^1 x^3 + C_k^1 x^2 + D_k^1 x + E_k^1)P_{k-4} + (F_k^1 x^3 + G_k^1 x^2 + H_k^1 x + I_k^1)P_{k-3}, \quad (2.99)$$

where $P_k^{(1)}(x)$, $P_{k-3}(x)$ and $P_{k-4}(x)$ are polynomials of degree $k$, $k-3$ and $k-4$ respectively. The constant coefficients $A_k^1$, $B_k^1$, $C_k^1$, $D_k^1$, $E_k^1$,$F_k^1$, $G_k^1$ and $H_k^1$ are determined. After multiplying equation (2.99) by $x^i$ and applying linear functional $c^{(1)}$ on both sides it becomes

$$\begin{aligned} c^{(1)}(x^i P_k) \quad = \quad & A_k^1 c(x^{i+5}P_{k-4}) + B_k^1 c(x^{i+4}P_{k-4}) + C_k^1 c(x^{i+3}P_{k-4}) + D_k^1 c(x^{i+2}P_{k-4}) + E_k^1 c(x^{i+1}P_{k-4}) \\ & + F_k^1 c(x^{i+4}P_{k-3}) + G_k^1 c(x^{i+3}P_{k-3}) + H_k^1 c(x^{i+2}P_{k-3}) + I_k^1 c(x^{i+1}P_{k-3}). \quad (2.100) \end{aligned}$$

Consequently, by applying (2.2), we have the relation for $i = 0, 1, ..., k-1$

$$\begin{aligned} & A_k^1 c(x^{i+5}P_{k-4}) + B_k^1 c(x^{i+4}P_{k-4}) + C_k^1 c(x^{i+3}P_{k-4}) + D_k^1 c(x^{i+2}P_{k-4}) + E_k^1 c(x^{i+1}P_{k-4}) + \\ & F_k^1 c(x^{i+4}P_{k-3}) + G_k^1 c(x^{i+3}P_{k-3}) + H_k^1 c(x^{i+2}P_{k-3}) + I_k^1 c(x^{i+1}P_{k-3}) = 0. \quad (2.101) \end{aligned}$$

The orthogonality condition is always true for $i = 0, 1, 2, ......, k-10$. Therefore for $i = k-9$, equation (2.101) gives

$$A_k^1 c(x^{k-4}P_{k-4}) = 0, \Rightarrow c(x^{k-4}P_{k-4}) \neq 0, \; A_k^1 = 0.$$

For $i = k-8$, equation (2.101) gives

$$B_k^1 c(x^{k-4}P_{k-4}) = 0, \Rightarrow c(x^{k-4}P_{k-4}) \neq 0, \; B_k^1 = 0.$$

For $i = k-7$, equation (2.101) gives

$$C_k^1 c(x^{k-4}P_{k-4}) + F_k^1 c(x^{k-3}P_{k-3}) = 0. \quad (2.102)$$

For $i = k-6$, equation (2.101) gives

$$C_k^1 c(x^{k-3}P_{k-4}) + D_k^1 c(x^{k-4}P_{k-4}) + F_k^1 c(x^{k-2}P_{k-3}) + G_k^1 c(x^{k-3}P_{k-3}) = 0. \tag{2.103}$$

For $i = k-5$, equation (2.101) gives

$$C_k^1 c(x^{k-2}P_{k-4}) + D_k^1 c(x^{k-3}P_{k-4}) + E_k^1 c(x^{k-4}P_{k-4}) + F_k^1 c(x^{k-1}P_{k-3}) + G_k^1 c(x^{k-2}P_{k-3}) + H_k^1 c(x^{k-3}P_{k-3}) = 0. \tag{2.104}$$

For $i = k-4$, equation (2.101) gives

$$C_k^1 c(x^{k-1}P_{k-4}) + D_k^1 c(x^{k-2}P_{k-4}) + E_k^1 c(x^{k-3}P_{k-4}) + F_k^1 c(x^{k}P_{k-3}) + G_k^1 c(x^{k-1}P_{k-3}) + H_k^1 c(x^{k-2}P_{k-3}) + I_k^1 c(x^{k-3}P_{k-3}) = 0. \tag{2.105}$$

For $i = k-3$, equation (2.101) gives

$$C_k^1 c(x^{k}P_{k-4}) + D_k^1 c(x^{k-1}P_{k-4}) + E_k^1 c(x^{k-2}P_{k-4}) + F_k^1 c(x^{k+1}P_{k-3}) + G_k^1 c(x^{k}P_{k-3}) + H_k^1 c(x^{k-1}P_{k-3}) + I_k^1 c(x^{k-2}P_{k-3}) = 0. \tag{2.106}$$

For $i = k-2$, equation (2.101) gives

$$C_k^1 c(x^{k+1}P_{k-4}) + D_k^1 c(x^{k}P_{k-4}) + E_k^1 c(x^{k-1}P_{k-4}) + F_k^1 c(x^{k+2}P_{k-3}) + G_k^1 c(x^{k+1}P_{k-3}) + H_k^1 c(x^{k}P_{k-3}) + I_k^1 c(x^{k-1}P_{k-3}) = 0. \tag{2.107}$$

For $i = k-1$, equation (2.101) gives

$$C_k^1 c(x^{k+2}P_{k-4}) + D_k^1 c(x^{k+1}P_{k-4}) + E_k^1 c(x^{k}P_{k-4}) + F_k^1 c(x^{k+3}P_{k-3}) + G_k^1 c(x^{k+1}P_{k-3}) + H_k^1 c(x^{k+1}P_{k-3}) + I_k^1 c(x^{k}P_{k-3}) = 0. \tag{2.108}$$

Since the above system of equations is homogenous. Its coefficient matrix is non-singular, we get $C_k^1 = D_k^1 = E_k^1 = F_k^1 = G_k^1 = H_k^1 = I_k^1 = 0$.

Hence the recurrence relation $B_{18}$ becomes

$$P_k^{(1)} = 0.$$

Hence, a Lanczos-type algorithm based on $B_{18}$ cannot be implemented.

### 2.3.3 $B_{19}$ for $U_i(x) = x^i$

Consider the following recurrence relation for $k \geq 4$,

$$P_k^{(1)}(x) = (A_k^1 x^4 + B_k^1 x^3 + C_k^1 x^2 + D_k^1 x + E_k^1)P_{k-4}^{(1)} + (F_k^1 x^3 + G_k^1 x^2 + H_k^1 x + I_k^1)P_{k-3}^{(1)}. \quad (2.109)$$

where $P_k^{(1)}$, $P_{k-3}^{(1)}$ and $P_{k-4}^{(1)}$ be the orthogonal polynomials of degree $k$, $k-3$ and $k-4$ respectively. The constant coefficients $A_k^1$, $B_k^1$, $C_k^1$, $D_k^1$, $E_k^1$, $F_k^1$, $G_k^1$, $H_k^1$ and $I_k^1$ are to be determined. After multiplying equation (2.109) by $x^i$ and applying $c^{(1)}$ on both sides it becomes

$$\begin{aligned} c^{(1)}(x^i P_k^{(1)}) &= A_k^1 c^{(1)}(x^{i+4} P_{k-4}^{(1)}) + B_k^1 c^{(1)}(x^{i+3} P_{k-4}^{(1)}) + C_k^1 c^{(1)}(x^{i+2} P_{k-4}^{(1)}) + D_k^1 c^{(1)}(x^{i+1} P_{k-4}^{(1)}) + \\ &\quad E_k^1 c^{(1)}(x^i P_{k-4}^{(1)}) + F_k^1 c^{(1)}(x^{i+3} P_{k-3}^{(1)}) + G_k^1 c^{(1)}(x^{i+2} P_{k-3}^{(1)}) + H_k^1 c^{(1)}(x^{i+1} P_{k-3}^{(1)}) + I_k^1 c^{(1)}(x^i P_{k-3}^{(1)}). \end{aligned}$$

Consequently, by applying (2.2), we have the relation for $i = 0, 1, ..., k-1$

$$\begin{aligned} &A_k^1 c^{(1)}(x^{i+4} P_{k-4}^{(1)}) + B_k^1 c^{(1)}(x^{i+3} P_{k-4}^{(1)}) + C_k^1 c^{(1)}(x^{i+2} P_{k-4}^{(1)}) + D_k^1 c^{(1)}(x^{i+1} P_{k-4}^{(1)}) + E_k^1 c^{(1)}(x^i P_{k-4}^{(1)}) + \\ &\quad F_k^1 c^{(1)}(x^{i+3} P_{k-3}^{(1)}) + G_k^1 c^{(1)}(x^{i+2} P_{k-3}^{(1)}) + H_k^1 c^{(1)}(x^{i+1} P_{k-3}^{(1)}) + I_k^1 c^{(1)}(x^i P_{k-3}^{(1)}) = 0. \quad (2.110) \end{aligned}$$

For $i = 0, 1, 2, ..., k-9$, the relation (2.110) is always true.

Therefore, for $i = k-8$, equation (2.110) gives

$$A_k^1 c^{(1)}(x^{k-4} P_{k-4}^{(1)}) = 0 \;\Rightarrow\; c^{(1)}(x^{k-4} P_{k-4}^{(1)}) \neq 0,\; A_k^1 = 0.$$

Since $P_k^{(1)}(x)$ is monic-polynomial of degree $k$, therefore, $F_k^1 = 1$.

For $i = k-7$, equation (2.110) gives

$$B_k^1 c^{(1)}(x^{k-4} P_{k-4}^{(1)}) = 0 \Rightarrow c^{(1)}(x^{k-4} P_{k-4}^{(1)}) \neq 0, \quad B_k^1 = 0.$$

For $i = k-6$, equation (2.110) gives

$$C_k^1 = -\frac{c(x^{k-2} P_{k-3}^{(1)})}{c(x^{k-3} P_{k-4}^{(1)})}. \quad (2.111)$$

For $i = k-5$, equation (2.110) gives

$$D_k^1 c^{(1)}(x^{k-4} P_{k-4}^{(1)}) + G_k^1 c^{(1)}(x^{k-3} P_{k-3}^{(1)}) = -c^{(1)}(x^{k-2} P_{k-3}^{(1)}) - C_k^1 c^{(1)}(x^{k-3} P_{k-4}^{(1)}). \quad (2.112)$$

For $i = k-4$, equation (2.110) gives

$$D_k^1 c^{(1)}(x^{k-3}P_{k-4}^{(1)}) + E_k^1 c^{(1)}(x^{k-4}P_{k-4}^{(1)}) + G_k^1 c^{(1)}(x^{k-2}P_{k-3}^{(1)}) + H_k^1 c^{(1)}(x^{k-3}P_{k-3}^{(1)}) = -c^{(1)}(x^{k-1}P_{k-3}^{(1)}) - C_k^1 c^{(1)}(x^{k-2}P_{k-4}^{(1)}). \quad (2.113)$$

For $i = k-3$, equation (2.110) gives

$$D_k^1 c^{(1)}(x^{k-2}P_{k-4}^{(1)}) + E_k^1 c^{(1)}(x^{k-3}P_{k-4}^{(1)}) + G_k^1 c^{(1)}(x^{k-1}P_{k-3}^{(1)}) + H_k^1 c^{(1)}(x^{k-2}P_{k-3}^{(1)}) + I_k^1 c^{(1)}(x^{k-3}P_{k-3}^{(1)}) = -c^{(1)}(x^{k}P_{k-3}^{(1)}) - C_k^1 c^{(1)}(x^{k-1}P_{k-4}^{(1)}). \quad (2.114)$$

For $i = k-2$, equation (2.110) gives

$$D_k^1 c^{(1)}(x^{k-1}P_{k-4}^{(1)}) + E_k^1 c^{(1)}(x^{k-2}P_{k-4}^{(1)}) + G_k^1 c^{(1)}(x^{k}P_{k-3}^{(1)}) + H_k^1 c^{(1)}(x^{k-1}P_{k-3}^{(1)}) + I_k^1 c^{(1)}(x^{k-2}P_{k-3}^{(1)}) = -c^{(1)}(x^{k+1}P_{k-3}^{(1)}) - C_k^1 c^{(1)}(x^{k}P_{k-4}^{(1)}). \quad (2.115)$$

For $i = k-1$, equation (2.110) gives

$$D_k^1 c^{(1)}(x^{k}P_{k-4}^{(1)}) + E_k^1 c^{(1)}(x^{k-1}P_{k-4}^{(1)}) + G_k^1 c^{(1)}(x^{k+1}P_{k-3}^{(1)}) + H_k^1 c^{(1)}(x^{k}P_{k-3}^{(1)}) + I_k^1 c^{(1)}(x^{k-1}P_{k-3}^{(1)}) = -c^{(1)}(x^{k+2}P_{k-3}^{(1)}) - C_k^1 c^{(1)}(x^{k+1}P_{k-4}^{(1)}). \quad (2.116)$$

Equations (2.112), (2.113), (2.114), (2.115) and (2.116) can be written as

$$\begin{cases} a_{11}D_k^1 + a_{13}G_k^1 = b_1, \\ a_{21}D_k^1 + a_{22}E_k^1 + a_{23}G_k^1 + a_{24}H_k^1 = b_2, \\ a_{31}D_k^1 + a_{32}E_k^1 + a_{33}G_k^1 + a_{34}H_k^1 + a_{35}I_k^1 = b_3, \\ a_{41}D_k^1 + a_{42}E_k^1 + a_{43}G_k^1 + a_{44}H_k^1 + a_{45}I_k^1 = b_4, \\ a_{51}D_k^1 + a_{52}E_k^1 + a_{53}G_k^1 + a_{54}H_k^1 + a_{55}I_k^1 = b_5. \end{cases} \quad (2.117)$$

Where $a_{11}, a_{13}, a_{21}, a_{22}, a_{23}, a_{24}, a_{31}, a_{32}, a_{33}, a_{34}, a_{35}, a_{41}, a_{42}, a_{43}, a_{44}, a_{45}, a_{51}, a_{52}, a_{53}, a_{54}$, and $a_{55}$ are the coefficients of $D_k^1, E_k^1, G_k^1, H_k^1$ and $I_k^1$ respectively. Suppose $b_1, b_2, b_3, b_4$, and $b_5$ are the corresponding right hand side terms of these equations. If $\Delta_k$ represents the determinant of the coefficients matrix of (2.117) then we have,

$$\Delta_k = det(L), \tag{2.118}$$

where $L = matrix([l_1, l_2, l_3, l_4, l_5])$,

$l_1 = [a_{11}, 0, a_{13}, 0, 0],\ l_2 = [a_{21}, a_{22}, a_{23}, a_{24}, 0],\ l_3 = [a_{31}, a_{32}, a_{33}, a_{34}, a_{35}]$,

$l_4 = [a_{41}, a_{42}, a_{43}, a_{44}, a_{45}],\ l_5 = [a_{51}, a_{52}, a_{53}, a_{54}, a_{55}]$.

If $\Delta_k \neq 0$, then

$$\begin{cases} D_k^1 = \frac{det(M)}{\Delta_k}, \quad \text{where } M = matrix([m_1, m_2, m_3, m_4, m_5]), \\ m_1 = [b_1, 0, a_{13}, 0, 0],\ m_2 = [b_2, a_{22}, a_{23}, a_{24}, 0],\ m_3 = [b_3, a_{32}, a_{33}, a_{34}, a_{35}], \\ m_4 = [b_4, a_{42}, a_{43}, a_{44}, a_{45}],\ m_5 = [b_5, a_{52}, a_{53}, a_{54}, a_{55}], \\ E_k^1 = \frac{det(N)}{\Delta_k}, \quad \text{where } N = matrix([n_1, n_2, n_3, n_4, n_5]), \\ n_1 = [a_{11}, b_1, a_{13}, 0, 0],\ n_2 = [a_{21}, b_2, a_{23}, a_{24}, 0],\ n_3 = [a_{31}, b_3, a_{33}, a_{34}, a_{35}], \\ n_4 = [a_{41}, b_4, a_{43}, a_{44}, a_{45}],\ n_5 = [a_{51}, b_5, a_{53}, a_{54}, a_{55}], \\ G_k^1 = \frac{b_1 - a_{11}D_k}{a_{13}}, \\ H_k^1 = \frac{b_2 - a_{21}D_k - a_{22}E_k - a_{23}G_k}{a_{24}}, \\ I_k^1 = \frac{b_3 - a_{31}D_k - a_{32}E_k - a_{33}G_k - a_{34}H_k}{a_{35}}. \end{cases} \tag{2.119}$$

Since $A_k^1 = B_k^1 = 0$ and $F_k^1 = 1$, relation $B_{19}$ becomes

$$P_k^{(1)}(x) = \{C_k^1 x^2 + D_k^1 x + E_k^1) P_{k-4}^{(1)}(x) + (x^3 + G_k^1 x^2 + H_k^1 x + I_k^1) P_{k-3}^{(1)}(x)\}. \tag{2.120}$$

This means $B_{19}$ can lead to the implementation of a Lanczos-type algorithm.

### 2.3.4 $B_{20}$ for $U_i(x) = x^i$

Consider the following recurrence relationship for $k \geq 4$

$$P_k^{(1)}(x) = (A_k x^4 + B_k x^3 + C_k x^2 + D_k x + E_k) P_{k-4}^{(1)} + (F_k x^2 + G_k x + H_k) P_{k-2}, \tag{2.121}$$

where $P_k^{(1)}$, $P_{k-2}$ and $P_{k-4}^{(1)}$ are polynomials of degree $k$, $k-2$ and $k-4$ respectively. The constant coefficients $A_k^1$, $B_k^1$, $C_k^1$, $D_k^1$, $E_k^1$ $F_k^1$ $G_k^1$ and $H_k^1$ are determined. After multiplying equation (2.121) by $x^i$ and applying linear functional $c^{(1)}$ on both sides it becomes

$$c^{(1)}(x^iP_k) = A_k^1c^{(1)}(x^{i+4}P_{k-4}^{(1)}) + B_k^1c^{(1)}(x^{i+3}P_{k-4}^{(1)}) + C_k^1c^{(1)}(x^{i+2}P_{k-4}^{(1)}) + D_k^1c^{(1)}(x^{i+1}P_{k-4}^{(1)}) + E_k^1c^{(1)}(x^iP_{k-4}^{(1)})$$
$$+F_k^1c(x^{i+3}P_{k-2}) + G_k^1c(x^{i+2}P_{k-2}) + H_k^1c(x^{i+1}P_{k-2}).$$

Consequently, by applying (2.2), we have the relation for $i = 0, 1, ..., k-1$

$$A_k^1c^{(1)}(x^{i+4}P_{k-4}^{(1)}) + B_k^1c^{(1)}(x^{i+3}P_{k-4}^{(1)}) + C_k^1c^{(1)}(x^{i+2}P_{k-4}^{(1)}) + D_k^1c^{(1)}(x^{i+1}P_{k-4}^{(1)}) + E_k^1c^{(1)}(x^iP_{k-4}^{(1)})$$
$$+F_k^1c(x^{i+3}P_{k-2}) + G_k^1c(x^{i+2}P_{k-2}) + H_k^1c(x^{i+1}P_{k-2}) = 0. \quad (2.122)$$

For $i = 0, 1, 2, ..., k-9$, the relation (2.122) is always true. Therefore for $i = k-8$, equation (2.122) gives

$$A_k^1c^{(1)}(x^{k-4}P_{k-4}^{(1)}) = 0 \;\Rightarrow\; c^{(1)}(x^{k-4}P_{k-4}^{(1)}) \neq 0,\; A_k^1 = 0.$$

For $i = k-7$, $i = k-6$, equation (2.122) gives $B_k^1 = 0$, $C_k^1 = 0$ respectively. For $i = k-5$, $i = k-4$, $i = k-3$, $i = k-2$ and $i = k-1$. We get five homogenous equations and its coefficient matrix is non-singular, so we have $D_k^1 = E_k^1 = F_k^1 = G_k^1 = H_k^1 = 0$. This shows that the relation $B_{20}$ defined above becomes

$$P_k^{(1)} = 0.$$

Hence, a Lanczos-type algorithm based on $B_{20}$ cannot be implemented.

### 2.3.5 $B_{21}$ for $U_i(x) = x^i$

Let $P_k^{(1)}$, $P_{k-3}$ and $P_{k-3}^{(1)}$ be the orthogonal polynomials of degree $k, k-3$ and $k-3$ respectively and consider the following recurrence relation for $k \geq 3$,

$$P_k^{(1)}(x) = (A_k^1x^3 + B_k^1x^2 + C_k^1x + D_k^1)P_{k-3} + (E_k^1x^3 + F_k^1x^2 + G_k^1x + H_k^1)P_{k-3}^{(1)}, \quad (2.123)$$

The constant coefficients $A_k^1$, $B_k^1$, $C_k^1$, $D_k^1$, $E_k^1$, $F_k^1$ and $G_k^1$ are to be determined. After multiplying equation (2.123) by $x^i$ and applying linear function $c^{(1)}$ on both sides it becomes

$$\begin{aligned} c^{(1)}(x^i P_k^{(1)}) &= A_k^1 c^{(1)}(x^{i+3}P_{k-3}) + B_k^1 c^{(1)}(x^{i+2}P_{k-3}) + C_k^1 c^{(1)}(x^{i+1}P_{k-3}) + D_k^1 c^{(1)}(x^i P_{k-3}) + \\ & E_k^1 c^{(1)}(x^{i+3}P_{k-3}^{(1)}) + F_k^1 c^{(1)}(x^{i+2}P_{k-3}^{(1)}) + G_k^1 c^{(1)}(x^{i+1}P_{k-3}^{(1)}) + H_k^1 c^{(1)}(x^i P_{k-3}^{(1)}). \end{aligned}$$

Consequently, by applying (2.2), we have the relation for $i = 0, 1, ..., k-1$

$$\begin{aligned} A_k^1 c(x^{i+4}P_{k-3}) + B_k^1 c(x^{i+3}P_{k-3}) + C_k^1 c(x^{i+2}P_{k-3}) + D_k^1 c(x^{i+1}P_{k-3}) + E_k^1 c^{(1)}(x^{i+2}P_{k-3}^{(1)}) \\ + F_k^1 c^{(1)}(x^{i+1}P_{k-3}^{(1)}) + G_k^1 c^{(1)}(x^i P_{k-3}^{(1)}) + H_k^1 c(x^i P_{k-3}^{(1)}) = 0. \end{aligned} \tag{2.124}$$

For $i = 0, 1, 2, ..., k-8$, the relation (2.124) is always true. Therefore for $i = k-7$, equation (2.124) gives, $A_k^1 c(x^{k-3}P_{k-3}) = 0 \Rightarrow c(x^{k-3}P_{k-3}) \neq 0, \quad A_k^1 = 0$. Since $P_k^{(1)}(x)$ is monic, therefore $E_k^1 = 1$. For $i = k-6$, equation (2.124) gives

$$B_k^1 = -\frac{c(x^{k-2}P_{k-3}^{(1)})}{c(x^{k-3}P_{k-3})}. \tag{2.125}$$

For $i = k-5$, equation (2.124) gives

$$C_k^1 c(x^{k-3}P_{k-3}) + F_k^1 c^{(1)}(x^{k-3}P_{k-3}^{(1)}) = -c^{(1)}(x^{k-2}P_{k-3}^{(1)}) - B_k^1 c(x^{k-2}P_{k-3}). \tag{2.126}$$

For $i = k-4$, equation (2.124) gives

$$C_k^1 c(x^{k-2}P_{k-3}) + D_k^1 c(x^{k-3}P_{k-3}) + F_k^1 c^{(1)}(x^{k-2}P_{k-3}^{(1)}) + G_k^1 c^{(1)}(x^{k-3}P_{k-3}^{(1)}) = -c^{(1)}(x^{k-1}P_{k-3}^{(1)}) - B_k^1 c(x^{k-1}P_{k-3}). \tag{2.127}$$

For $i = k-3$, equation (2.124) gives

$$\begin{aligned} C_k^1 c(x^{k-1}P_{k-3}) + D_k^1 c(x^{k-2}P_{k-3}) + F_k^1 c^{(1)}(x^{k-1}P_{k-3}^{(1)}) + G_k^1 c^{(1)}(x^{k-2}P_{k-3}^{(1)}) + H_k^1 c^{(1)}(x^{k-3}P_{k-3}^{(1)}) \\ = -c^{(1)}(x^k P_{k-3}^{(1)}) - B_k^1 c(x^k P_{k-3}). \end{aligned} \tag{2.128}$$

For $i = k-2$, equation (2.124) gives

$$\begin{aligned} C_k^1 c(x^k P_{k-3}) + D_k^1 c(x^{k-1}P_{k-3}) + F_k^1 c^{(1)}(x^k P_{k-3}^{(1)}) + G_k^1 c^{(1)}(x^{k-1}P_{k-3}^{(1)}) + H_k^1 c^{(1)}(x^{k-2}P_{k-3}^{(1)}) \\ = -c^{(1)}(x^{k+1}P_{k-3}^{(1)}) - B_k^1 c(x^{k+1}P_{k-3}). \end{aligned} \tag{2.129}$$

For $i = k-1$, equation (2.124) gives

$$C_k^1 c(x^{k+1}P_{k-3}) + D_k^1 c(x^k P_{k-3}) + F_k^1 c^{(1)}(x^{k+1}P_{k-3}^{(1)}) + G_k^1 c^{(1)}(x^k P_{k-3}^{(1)}) + H_k^1 c^{(1)}(x^{k-1}P_{k-3}^{(1)})$$
$$= -c^{(1)}(x^{k+2}P_{k-3}^{(1)}) - B_k^1 c(x^{k+2}P_{k-3}). \qquad (2.130)$$

Equations (2.126), (2.127), (2.128), (2.129) and (2.130) can be written as

$$\begin{cases} a_{11}C_k^1 + a_{13}F_k^1 = b_1, \\ a_{21}C_k^1 + a_{22}D_k^1 + a_{23}F_k^1 + a_{24}G_k^1 = b_2, \\ a_{31}C_k^1 + a_{32}D_k^1 + a_{33}F_k^1 + a_{34}G_k^1 + H_k^1 a_{35} = b_3, \\ a_{41}C_k^1 + a_{42}D_k^1 + a_{43}F_k^1 + a_{44}G_k^1 + H_k^1 a_{45} = b_4, \\ a_{51}C_k^1 + a_{52}D_k^1 + a_{53}F_k^1 + a_{54}G_k^1 + H_k^1 a_{55} = b_5. \end{cases} \qquad (2.131)$$

Where $a_{11}$, $a_{13}$, $a_{21}$, $a_{22}$, $a_{23}$, $a_{24}$, $a_{31}$, $a_{32}$, $a_{33}$, $a_{34}$, $a_{35}$, $a_{41}$, $a_{42}$, $a_{43}$, $a_{44}$, $a_{45}$, $a_{51}$, $a_{52}$, $a_{53}$, $a_{54}$, and $a_{55}$ are the coefficients of $C_k^1$, $D_k^1$, $F_k^1$, $G_k^1$ and $H_k^1$ respectively. Suppose $b_1$, $b_2$, $b_3$, $b_4$, and $b_5$ are the corresponding right hand side terms of these equations. If $\Delta_k$ represents the determinant of the coefficients matrix of (2.131). From (2.118), if $\Delta_k \neq 0$, then

$$\begin{cases} C_k^1 = D_k^1 \text{ as in (2.119)}, \\ D_k^1 = E_k^1 \text{ as in (2.119)}, \\ F_k^1 = \frac{b_1 - a_{11}C_k^1}{a_{13}}, \\ G_k^1 = -\frac{b_2 - a_{21}C_k^1 - a_{22}D_k^1 - a_{23}F_k^1}{a_{24}}, \\ H_k^1 = \frac{b_3 - a_{31}C_k^1 - a_{32}D_k^1 - a_{33}F_k^1 - a_{34}G_k^1}{a_{35}}. \end{cases} \qquad (2.132)$$

Since $A_k^1 = 0$ and $E_k^1 = 1$, relation $B_{21}$ becomes

$$P_k^{(1)}(x) = \{B_k^1 x^2 + C_k^1 x + D_k^1)P_{k-3}(x) + (x^3 + F_k^1 x^2 + G_k^1 x + H_k^1)P_{k-3}^{(1)}(x)\}. \qquad (2.133)$$

Therefore, $B_{21}$ leads to a Lanczos-type algorithm.

## 2.4 Design of Lanczos-type Algorithms

In sections 2.2 and 2.3, we derived some new FOPs based recurrence relations. Here, we will derive new variants of the Lanczos algorithm based on these relations. By writing $\mathbf{r}_k = P_k(A)\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_k$ and $\mathbf{z}_k = P_k^{(1)}(A)\mathbf{r}_0$, the relations $A_i$ allow to derive expressions for $\mathbf{r}_k$ and $\mathbf{x}_k$, and the relations $B_j$ allow to find the expression of $\mathbf{z}_k$, recursively. Hence, new Lanczos-type algorithms are introduced.

### 2.4.1 Lanczos-type Algorithm Based on $A_{20}$

From the recurrence relation $A_{20}$ of subsection 2.2.1, the equation (2.16), after replacing $x$ by $A$. Since $\mathbf{r}_k = P_k(A)\mathbf{r}_0$, we have

$$\mathbf{r}_k = A_k\Big\{(A^3 + B_kA^2 + C_kA + D_k)\mathbf{r}_{k-3} + (G_kA^2 + H_kA + I_k)\mathbf{r}_{k-4}\Big\}. \tag{2.134}$$

Using $\mathbf{r}_k = b - A\mathbf{x}_k$, we get

$$\mathbf{x}_k = A_k\Big\{I_k\mathbf{x}_{k-4} + D_k\mathbf{x}_{k-3} - (A^2 + B_kA + C_k)\mathbf{r}_{k-3} - (G_kA + H_k)\mathbf{r}_{k-4}\Big\}. \tag{2.135}$$

Equations (2.134) and (2.135) define a Lanczos-type algorithm. Now, we have to find the expressions of the coefficients $A_k$, $B_k$, $C_k$, $D_k$, $G_k$, $H_k$ and $I_k$ appearing in them, have been derived in subsection (2.2.1). We know that

$$\begin{cases} c(x^kP_k) = ((A^T)^k\mathbf{y}, P_k(A)\mathbf{r}_0) = (\mathbf{y}_k, \mathbf{r}_k) \\ \text{with } \mathbf{y}_k = A^T\mathbf{y}_{k-1} \end{cases} \tag{2.136}$$

Therefore, we can write using Eq (2.136) we get

$$G_k = -\frac{(\mathbf{y}_{k-3}, \mathbf{r}_{k-3})}{(\mathbf{y}_{k-4}, \mathbf{r}_{k-4})}. \tag{2.137}$$

The rest of the coefficents can be written explicitly as follows;

$a_{11} = (\mathbf{y}_{k-3}, \mathbf{r}_{k-3})$, $a_{14} = (\mathbf{y}_{k-4}, \mathbf{r}_{k-4})$, $a_{21} = (\mathbf{y}_{k-2}, \mathbf{r}_{k-3})$, $a_{22} = a_{11}$, $a_{24} = (\mathbf{y}_{k-3}, \mathbf{r}_{k-4})$, $a_{25} = a_{14}$,

$a_{31} = (\mathbf{y}_{k-1}, \mathbf{r}_{k-3})$, $a_{32} = a_{21}$, $a_{33} = a_{11}$, $a_{34} = (\mathbf{y}_{k-2}, \mathbf{r}_{k-4})$, $a_{35} = a_{24}$, $a_{41} = (\mathbf{y}_k, \mathbf{r}_{k-3})$, $a_{42} = a_{31}$,

$a_{43} = a_{21}$, $a_{44} = (\mathbf{y}_{k-1}, \mathbf{r}_{k-4})$, $a_{45} = a_{34}$, $a_{51} = (\mathbf{y}_{k+1}, \mathbf{r}_{k-3})$, $a_{52} = a_{41}$, $a_{53} = a_{31}$, $a_{54} = (\mathbf{y}_k, \mathbf{r}_{k-4})$,

$a_{55} = a_{44}$.

Using these relations we get

$b_1 = -(\mathbf{y}_{k-2}, \mathbf{r}_{k-3}) - G_k(\mathbf{y}_{k-3}, \mathbf{r}_{k-4}) = -a_{21} - G_k a_{24}$,

$b_2 = -(\mathbf{y}_{k-1}, \mathbf{r}_{k-3}) - G_k(\mathbf{y}_{k-2}, \mathbf{r}_{k-4}) = -a_{31} - G_k a_{34}$,

$b_3 = -(\mathbf{y}_k, \mathbf{r}_{k-3}) - G_k(\mathbf{y}_{k-1}, \mathbf{r}_{k-4}) = -a_{41} - G_k a_{44}$,

$b_4 = -(\mathbf{y}_{k+1}, \mathbf{r}_{k-3}) - G_k(\mathbf{y}_k, \mathbf{r}_{k-4}) = -a_{51} - G_k a_{54}$,

$b_5 = -c(x^{k+2}P_{k-3}) - G_k c(x^{k+1}P_{k-4}) = -s - G_k t$,

where $s = c(x^{k+2}P_{k-3}) = (\mathbf{y}_{k+2}, \mathbf{r}_{k-3})$, $t = c(x^{k+1}P_{k-4}) = (\mathbf{y}_{k+1}, \mathbf{r}_{k-4})$.

Since all previous formulae are valid for $k \geq 4$, therefor we need $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$, $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$, which are necessary to evaluate (2.134) and (2.135) recursively, which are below.

Since $\mathbf{r}_k = P_k(A)\mathbf{r}_0$, therefore, we can write using (1.11), we get

$$\begin{cases} \mathbf{r}_1 = \mathbf{r}_0 - \frac{c_0}{c_1} A\mathbf{r}_0, \\ \mathbf{x}_1 = \mathbf{x}_0 + \frac{c_0}{c_1}\mathbf{r}_0. \end{cases} \tag{2.138}$$

Where $c_i = (\mathbf{y}, A^i\mathbf{r}_0)$. Again using (1.11), we get

$$\begin{cases} \mathbf{r}_2 = \mathbf{r}_0 - \alpha A\mathbf{r}_0 + \beta A^2\mathbf{r}_0, \\ \mathbf{x}_2 = \mathbf{x}_0 + \alpha\mathbf{r}_0 - \beta A\mathbf{r}_0, \end{cases} \tag{2.139}$$

with $\alpha = \frac{c_0c_3 - c_1c_2}{\rho}$, $\beta = \frac{c_0c_2 - c_1^2}{\rho}$ and $\rho = c_1c_3 - c_2^2$.

Again using (1.11), we get

$$\begin{cases} \mathbf{r}_3 = \mathbf{r}_0 - \eta A\mathbf{r}_0 + \mu A^2\mathbf{r}_0 - \nu A^3\mathbf{r}_0, \\ \mathbf{x}_3 = \mathbf{x}_0 + \eta\mathbf{r}_0 - \mu A\mathbf{r}_0 + \nu A^2\mathbf{r}_0. \end{cases} \tag{2.140}$$

Where

$$\eta = \frac{c_0(c_3c_5 - c_4^2) - c_2(c_1c_5 - c_2c_4) + c_3(c_1c_4 - c_2c_3)}{\omega},$$

$$\mu = \frac{c_0(c_2c_5 - c_3c_4) - c_1(c_1c_5 - c_2c_4) + c_3(c_1c_3 - c_2^2)}{\omega},$$

$$\nu = \frac{c_0(c_2c_4 - c_3^2) - c_1(c_1c_4 - c_2c_3) + c_2(c_1c_3 - c_2^2)}{\omega},$$

with $\omega = c_1(c_3c_5 - c_4^2) - c_2(c_2c_5 - c_3c_4) + c_3(c_2c_4 - c_3^2)$.

We finally have the following algorithm after gathering together all these formulae.

**Algorithm 1** Lanczos-type Algorithm based on relation $A_{20}$

**Input:** $A$ an $n \times n$ matrix, $\mathbf{b}$ an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $||\mathbf{r}_k||$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, such that $\mathbf{y} \neq 0$ and the tolerance $\varepsilon$ to $1.0E-13$.
Set: $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$, $\mathbf{y}_0 = \mathbf{y}$.
**Compute:**
$c_0, c_1, c_2, c_3$, $c_4$ and $c_5$ as in (1.23b).
$\mathbf{r}_1$, $\mathbf{x}_1$, $\mathbf{r}_2$, $\mathbf{x}_2$, $\mathbf{r}_3$ and $\mathbf{x}_3$ as in (2.138), (2.139) and (2.140).
$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4$, $\mathbf{y}_5$ with $\mathbf{y}_k = A^T\mathbf{y}_{k-1}$.
$k = 4$,
**While** $||\mathbf{r}_k|| > \varepsilon$ **do**
$\mathbf{y}_{k+2} = A^T\mathbf{y}_{k+1}$,
$A_k$, as in (2.4),
$B_k$, $C_k$, $D_k$, $H_k$ and $I_k$, as in (2.15);
$G_k$, as in (2.137).
$\mathbf{r}_k = A_k\Big\{(A^3 + B_kA^2 + C_kA + D_k)\mathbf{r}_{k-3} + (G_kA^2 + H_kA + I_k)\mathbf{r}_{k-4}\Big\}$,
$\mathbf{x}_k = A_k\Big\{I_k\mathbf{x}_{k-4} + D_k\mathbf{x}_{k-3} - (A^2 + B_kA + C_k)\mathbf{r}_{k-3} - (G_kA + H_k)\mathbf{r}_{k-4}\Big\}$.
$k = k + 1$,
**EndWhile**
Obtain the approximate solution as well as the residual norm;
$\text{sol}_{last} = \mathbf{x}_k$,
$\text{norm}_{last} = ||\mathbf{r}_k||$.
**Stop**.

### 2.4.2 Lanczos-type Algorithm Based on $A_{22}/B_{19}$

From recurrence relation $A_{22}$ of subsection 2.2.3, the equation (2.33), after replacing $x$ by $A$.

Since $\mathbf{r}_k = P_k(A)\mathbf{r}_0$, we have

$$\mathbf{r}_k = \mathbf{r}_{k-3} + A_k\Big\{(A^3 + B_kA^2 + C_kA)\mathbf{r}_{k-3} + (F_kA^3 + G_kA^2 + H_kA)\mathbf{z}_{k-4}\Big\}. \quad (2.141)$$

$\because A_k D_k = 1$. Using $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, we get

$$\mathbf{x}_k = \mathbf{x}_{k-3} - A_k\Big\{(A^2 + B_k A + C_k)\mathbf{r}_{k-3} + (F_k A^2 + G_k A + H_k)\mathbf{z}_{k-4}\Big\}. \tag{2.142}$$

Equations (2.141) and (2.142) define a Lanczos-type algorithm. Now, we have to find the expressions of the coefficients $A_k$, $B_k$, $C_k$, $D_k$, $F_k$, $G_k$, and $H_k$ appearing in them, have been derived in subsection 2.2.3. Therefore, we can write using Eq (2.136) we get

$$F_k = -\frac{(\mathbf{y}_{k-3}, \mathbf{r}_{k-3})}{(\mathbf{y}_{k-3}, \mathbf{z}_{k-4})}. \tag{2.143}$$

The rest of the coefficents can be written explicitly as follow;

$a_{11} = (\mathbf{y}_{k-3}, \mathbf{r}_{k-3})$, $a_{14} = (\mathbf{y}_{k-3}, \mathbf{z}_{k-4})$, $a_{21} = (\mathbf{y}_{k-2}, \mathbf{r}_{k-3})$, $a_{22} = a_{11}$, $a_{24} = (\mathbf{y}_{k-2}, \mathbf{z}_{k-4})$, $a_{25} = a_{14}$,

$a_{31} = (\mathbf{y}_{k-1}, \mathbf{r}_{k-3})$, $a_{32} = a_{21}$, $a_{33} = a_{11}$, $a_{34} = (\mathbf{y}_{k-1}, \mathbf{z}_{k-4})$, $a_{35} = a_{24}$, $a_{41} = (\mathbf{y}_k, \mathbf{r}_{k-3})$, $a_{42} = a_{31}$,

$a_{43} = a_{21}$, $a_{44} = (\mathbf{y}_k, \mathbf{z}_{k-4})$, $a_{45} = a_{34}$, $a_{51} = (\mathbf{y}_{k+1}, \mathbf{r}_{k-3})$, $a_{52} = a_{41}$, $a_{53} = a_{31}$, $a_{54} = (\mathbf{y}_{k+1}, \mathbf{z}_{k-4})$,

$a_{55} = a_{44}$,

Using these relations we get

$b_1 = -(\mathbf{y}_{k-2}, \mathbf{r}_{k-3}) - F_k(\mathbf{y}_{k-2}, \mathbf{z}_{k-4}) = -a_{21} - F_k a_{24}$, $b_2 = -(\mathbf{y}_{k-1}, \mathbf{r}_{k-3}) - F_k(\mathbf{y}_{k-1}, \mathbf{z}_{k-4}) = -a_{31} - F_k a_{34}$,

$b_3 = -(\mathbf{y}_k, \mathbf{r}_{k-3}) - F_k(\mathbf{y}_k, \mathbf{z}_{k-4}) = -a_{41} - F_k a_{44}$, $b_4 = -(\mathbf{y}_{k+1}, \mathbf{r}_{k-3}) - F_k(\mathbf{y}_{k+1}, \mathbf{z}_{k-4}) = a_{51} - F_k a_{54}$,

$b_5 = -c(x^{k+2}P_{k-3}) - F_k c^{(1)}(x^{k+1}P^{(1)}_{k-4}) = -s - F_k t$,

$s = c(x^{k+2}P_{k-3}) = (\mathbf{y}_{k+2}, \mathbf{r}_{k-3})$, $t = c^{(1)}(x^{k+1}P^{(1)}_{k-4}) = (\mathbf{y}_{k+2}, \mathbf{z}_{k-4})$

All previous formulae are valid for $k \geq 4$, therefor we need $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$, $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$, which are necessary to evaluate (2.141) and (2.142) recursively, which can be computed by equations (2.138), (2.139) and (2.140) respectively.

From recurrence relation $B_{19}$ of subsection 2.3.3, the equation (2.120), after replacing $x$ by $A$. Since $\mathbf{z}_k = P^{(1)}_k(A)\mathbf{r}_0$, we have

$$\mathbf{z}_k = (C^1_k A^2 + D^1_k A + E^1_k)\mathbf{z}_{k-4} + (A^3 + G^1_k A^2 + H^1_k A + I^1_k)\mathbf{z}_{k-3}. \tag{2.144}$$

Now, we have to find the expressions of the coefficients $C_k^1$, $D_k^1$, $E_k^1$, $G_k^1$, $H_k^1$ and $I_k^1$ appearing in them, have been derived in subsection 2.3.3. Therefore, we can write using Eq (2.136) we get

$$C_k^1 = -\frac{(\mathbf{y}_{k-2}, \mathbf{z}_{k-3})}{(\mathbf{y}_{k-3}, \mathbf{z}_{k-4})}. \tag{2.145}$$

The rest of the coefficents can be written explicitly as follow;

$a_{11} = (\mathbf{y}_{k-3}, \mathbf{z}_{k-4})$, $a_{13} = (\mathbf{y}_{k-2}, \mathbf{z}_{k-3})$, $a_{21} = (\mathbf{y}_{k-2}, \mathbf{z}_{k-4})$, $a_{22} = a_{11}$, $a_{23} = (\mathbf{y}_{k-1}, \mathbf{z}_{k-3})$, $a_{24} = a_{13}$,

$a_{31} = (\mathbf{y}_{k-1}, \mathbf{z}_{k-4})$, $a_{32} = a_{21}$, $a_{33} = (\mathbf{y}_k, \mathbf{z}_{k-3})$, $a_{34} = a_{23}$, $a_{35} = a_{13}$, $a_{41} = (\mathbf{y}_k, \mathbf{z}_{k-4})$, $a_{42} = a_{31}$,

$a_{43} = (\mathbf{y}_{k+1}, \mathbf{z}_{k-3})$, $a_{44} = a_{33}$, $a_{45} = a_{23}$, $a_{51} = (\mathbf{y}_{k+1}, \mathbf{z}_{k-4})$, $a_{52} = a_{41}$, $a_{53} = (\mathbf{y}_{k+2}, \mathbf{z}_{k-3})$,

$a_{54} = a_{43}$, $a_{55} = a_{33}$,

Using these relations we get

$b_1 = -(\mathbf{y}_{k-1}, \mathbf{z}_{k-3}) - C_k^1(\mathbf{y}_{k-2}, \mathbf{z}_{k-4}) = -a_{23} - C_k^1 a_{21}$, $b_2 = -(\mathbf{y}_k, \mathbf{z}_{k-3}) - C_k^1(\mathbf{y}_{k-1}, \mathbf{z}_{k-4}) = -a_{33} - C_k^1 a_{31}$,

$b_3 = -(\mathbf{y}_{k+1}, \mathbf{z}_{k-3}) - C_k^1(\mathbf{y}_k, \mathbf{z}_{k-4}) = -a_{43} - C_k^1 a_{41}$, $b_4 = -(\mathbf{y}_{k+2}, \mathbf{z}_{k-3}) - C_k^1(\mathbf{y}_{k+1}, \mathbf{z}_{k-4}) = -a_{53} - C_k^1 a_{51}$,

$b_5 = -c^{(1)}(x^{k+2}P_{k-3}^{(1)}) - C_k^1 c^{(1)}(x^{k+1}P_{k-4}^{(1)}) = -s - C_k^1 t$

where $s = (\mathbf{y}_{k+3}, \mathbf{z}_{k-3})$ and $t = (\mathbf{y}_{k+2}, \mathbf{z}_{k-4})$ If $\Delta_k = 0$, then there is ghost-breakdown, [12, 18]. For $k \geq 4$, all above formulae are valid. This means that we have to find $\mathbf{z}_1$, $\mathbf{z}_2$ and $\mathbf{z}_3$ by alternative ways. Since $\mathbf{z}_k = P_k^{(1)}\mathbf{r}_0$,, therefore, we can write using (1.16), we get

$$\mathbf{z}_1 = A\mathbf{r}_0 - \frac{c_2}{c_1}\mathbf{r}_0. \tag{2.146}$$

Again from (1.16), we have

$$\mathbf{z}_2 = A^2\mathbf{r}_0 - \mu A\mathbf{r}_0 + \nu\mathbf{r}_0, \tag{2.147}$$

where $\mu = \frac{c_1c_4 - c_2c_3}{\rho}$, $\nu = \frac{c_2c_4 - c_3^2}{\rho}$ with $\rho = c_1c_3 - c_2^2$.

Similarly we have

$$\mathbf{z}_3 = A^3\mathbf{r}_0 - \eta' A^2\mathbf{r}_0 + \mu' A\mathbf{r}_0 - \nu'\mathbf{r}_0. \tag{2.148}$$

where

$$\eta' = \frac{c_1(c_3c_6 - c_4c_5) - c_2(c_2c_6 - c_3c_5) + c_4(c_2c_4 - c_3^2)}{\rho'},$$

$$\mu' = \frac{c_1(c_4c_6 - c_5^2) - c_3(c_2c_6 - c_3c_5) + c_4(c_2c_5 - c_4c_3)}{\rho'},$$

$$\nu' = \frac{c_2(c_4c_6 - c_5^2) - c_3(c_3c_6 - c_4c_5) + c_4(c_3c_5 - c_4^2)}{\rho'}.$$

with $\rho' = c_1(c_3c_5 - c_4^2) - c_2(c_2c_5 - c_3c_4) + c_3(c_2c_4 - c_3^2)$.

We finally have the following algorithm after gathering together all these formulae.

**Algorithm 2** Lanczos-type Algorithm based on relations $A_{22}/B_{19}$

**Input:** $A$ an $n \times n$ matrix, $\mathbf{b}$ an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $\|\mathbf{r}_k\|$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, such that $\mathbf{y} \neq 0$ and the tolerance $\varepsilon$ to $1.0E-13$.
Set $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$; $\mathbf{y}_0 = \mathbf{y}$; $\mathbf{z}_0 = \mathbf{r}_0$.
**Compute:**
$c_0, c_1, c_2, c_3, c_4$ and $c_5$; as in (1.23b).
$\mathbf{r}_1$, $\mathbf{x}_1$, $\mathbf{r}_2$, $\mathbf{x}_2$, $\mathbf{r}_3$ and $\mathbf{x}_3$ as in (2.138), (2.139) and (2.140).
$\mathbf{z}_1$, $\mathbf{z}_2$, and $\mathbf{z}_3$, as in (2.146), (2.147) and (2.148).
$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4$ with $\mathbf{y}_k = A^T\mathbf{y}_{k-1}$.
$k = 3$;
**While** $\|\mathbf{r}_k\| > \varepsilon$ **do**
$\mathbf{y}_{k+2} = A^T\mathbf{y}_{k+1}$.
$A_k$, as in (2.24),
$B_k$, $C_k$, $D_k$, $G_k$, and $H_k$, as in (2.32);
$F_k$ as in (2.143).
$\mathbf{r}_k = \mathbf{r}_{K-3} + A_k\Big\{(A^3 + B_kA^2 + C_kA)\mathbf{r}_{k-3} + (F_kA^3 + G_kA^2 + H_kA)\mathbf{z}_{k-4}\Big\}$,
$\mathbf{x}_k = \mathbf{x}_{k-3} - A_k\Big\{(A^2 + B_kA + C_k)\mathbf{r}_{k-3} + (F_kA^2 + G_k)A + H_k)\mathbf{z}_{k-4}\Big\}$.
$C_k^1$, as in (2.145);
$D_k^1$, $E_k^1$, $G_k^1$, $H_k^1$, and $I_k^1$ as in (2.119).
$\mathbf{z}_k = (C_k^1A^2 + D_k^1A + E_k^1)\mathbf{z}_{k-4} + (A^3 + G_k^1A^2 + H_k^1A + I_k^1)\mathbf{z}_{k-3}$.
$k = k + 1$,
**EndWhile**
Obtain the approximate solution as well as the residual norm;
$\text{sol}_{last} = \mathbf{x}_k$,
$\text{norm}_{last} = \|\mathbf{r}_k\|$.
**Stop**.

### 2.4.3 Lanczos-type Algorithm Based on $A_{22}/B_{21}$

The relation $A_{22}$ of this algorithm have already been derived in subsection 2.4.2. From Eqs (2.141) and (2.142) we have

$$\begin{cases} \mathbf{r}_k(x) = \mathbf{r}_{k-3} + A_k\Big\{(A^3 + B_kA^2 + C_kA)\mathbf{r}_{k-3} + (F_kA^3 + G_kA^2 + H_kA)\mathbf{z}_{k-4}(x)\Big\}, \\ \mathbf{x}_k = \mathbf{x}_{k-3} - A_k\Big\{(A^2 + B_kA + C_k)\mathbf{r}_{k-3} + (F_kA^2 + G_kA + H_k)\mathbf{z}_{k-4}\Big\}. \end{cases} \tag{2.149}$$

Equations (2.149) define a Lanczos-type algorithm. Now, we have to find the expressions of the coefficients $A_k$, $B_k$, $C_k$, $D_k$, $F_k$, $G_k$, and $H_k$ appearing in them, have been derived in subsection 2.4.2. Since all previous formulae are valid for $k \geq 4$, therefore we need $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$, $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$, which are necessary to evaluate (2.149) recursively, which are given as in equations (2.138), (2.139) and (2.140).

From relation $B_{21}$ of subsection 2.3.5, the Eq (2.133), after replacing $x$ by $A$.

Since $\mathbf{z}_k = P_k^{(1)}(A)\mathbf{r}_0$ we have

$$\mathbf{z}_k = (B_k^1A^2\mathbf{r}_{k-3} + C_k^1A\mathbf{r}_{k-3} + D_k^1\mathbf{r}_{k-3} + A^3\mathbf{z}_{k-3} + F_k^1A^2\mathbf{z}_{k-3} + G_k^1A\mathbf{z}_{k-3} + H_k^1\mathbf{z}_{k-3}). \tag{2.150}$$

Now, we have to find the expressions of the coefficients $B_k^1$, $C_k^1$, $D_k^1$, $F_k^1$, $G_k^1$, and $H_k^1$ appearing in them, have been derived in subsection 2.3.5. Therefore, we can write using Eq (2.136) we get

$$B_k^1 = -\frac{(\mathbf{y}_{k-2}, \mathbf{z}_{k-3})}{(\mathbf{y}_{k-3}, \mathbf{r}_{k-3})}. \tag{2.151}$$

The rest of the coefficents can be written explicitly as follow;

$a_{11} = (\mathbf{y}_{k-3}, \mathbf{r}_{k-3})$, $a_{13} = (\mathbf{y}_{k-2}, \mathbf{z}_{k-3})$, $a_{21} = (\mathbf{y}_{k-2}, \mathbf{r}_{k-3})$, $a_{22} = a_{11}$, $a_{23} = (\mathbf{y}_{k-1}, \mathbf{z}_{k-3})$, $a_{24} = a_{13}$,

$a_{31} = (\mathbf{y}_{k-1}, \mathbf{r}_{k-3})$, $a_{32} = a_{21}$, $a_{33} = (\mathbf{y}_k, \mathbf{z}_{k-3})$, $a_{34} = (\mathbf{y}_{k-1}, \mathbf{z}_{k-3})$, $a_{35} = a_{13}$, $a_{41} = (\mathbf{y}_k, \mathbf{r}_{k-3})$,

$a_{42} = a_{31}$, $a_{43} = (\mathbf{y}_{k+1}, \mathbf{z}_{k-3})$, $a_{44} = a_{33}$, $a_{45} = a_{23}$, $a_{51} = (\mathbf{y}_{k+1}, \mathbf{r}_{k-3})$, $a_{52} = a_{41}$, $a_{53} = (\mathbf{y}_{k+2}, \mathbf{z}_{k-3})$,

$a_{54} = a_{43}$, $a_{55} = a_{33}$

Using these relations we get

$b_1 = -(\mathbf{y}_{k-1}, \mathbf{z}_{k-3}) - B_k^1(\mathbf{y}_{k-2}, \mathbf{r}_{k-3}) = -a_{23} - B_k^1 a_{21}, \ b_2 = -(\mathbf{y}_k, \mathbf{z}_{k-3}) - B_k^1(\mathbf{y}_{k-1}, \mathbf{r}_{k-3}) = -a_{33} - B_k^1 a_{31}$

$b_3 = -(\mathbf{y}_{k+1}, \mathbf{z}_{k-3}) - B_k^1(\mathbf{y}_k, \mathbf{r}_{k-3}) = -a_{43} - B_k^1 a_{41}, \ b_4 = -(\mathbf{y}_{k+2}, \mathbf{z}_{k-3}) - B_k^1(\mathbf{y}_{k+1}, \mathbf{r}_{k-3}) = -a_{53} - B_k^1 a_{51}$

$b_5 = -(\mathbf{y}_{k+3}, \mathbf{z}_{k-3}) - B_k^1(\mathbf{y}_{k+2}, \mathbf{r}_{k-3}) = -s' - B_k^1 t',$

where $s' = (\mathbf{y}_{k+3}, \mathbf{z}_{k-3})$ and $t' = (\mathbf{y}_{k+2}, \mathbf{r}_{k-3})$.

If $\Delta_k = 0$, then there is ghost-breakdown, [12, 18]. For $k \geq 3$, all above formulae are valid.

This means that we have to find $\mathbf{z}_1$, $\mathbf{z}_2$ and $\mathbf{z}_3$ by alternative ways as in subsection 2.4.2.

Which can be computed by equations (2.146), (2.147) and (2.18).

We finally have the following algorithm after gathering together all these formulae.

**Algorithm 3** Lanczos-type Algorithm based on relations $A_{22}/B_{21}$

**Input:** $A$ an $n \times n$ matrix, $\mathbf{b}$ an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $\|\mathbf{r}_k\|$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, such that $\mathbf{y} \neq 0$ and the tolerance $\varepsilon$ to $1.0E - 13$.
Set $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$; $\mathbf{y}_0 = \mathbf{y}$; $\mathbf{z}_0 = \mathbf{r}_0$.
**Compute:**
$c_0, c_1, c_2, c_3, c_4$ and $c_5$; as in (1.23b),
$\mathbf{r}_1, \mathbf{x}_1, \mathbf{r}_2, \mathbf{x}_2, \mathbf{r}_3$ and $\mathbf{x}_3$ as in (2.138), (2.139) and (2.140),
$\mathbf{z}_1, \mathbf{z}_2,$ and $\mathbf{z}_3$, as in (2.146), (2.147) and (2.148),
$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4$ with $\mathbf{y}_k = A^T \mathbf{y}_{k-1}$.
$k = 3,$
**While** $\|\mathbf{r}_k\| > \varepsilon$ **do**
$\mathbf{y}_{k+2} = A^T \mathbf{y}_{k+1},$
$A_k$, as in (2.24), $B_k$, $C_k$, $D_k$, $G_k$, and $H_k$, as in (2.32) and $F_k$ as in (2.143),
$\mathbf{r}_k = \mathbf{r}_{k-3} + A_k\Big\{(A^3 + B_k A^2 + C_k A)\mathbf{r}_{k-3} + (F_k A^3 + G_k A^2 + H_k A)\mathbf{z}_{k-4}\Big\},$
$\mathbf{x}_k = \mathbf{x}_{k-3} - A_k\Big\{(A^2 + B_k A + C_k)\mathbf{r}_{k-3} + (F_k A^2 + G_k)A + H_k)\mathbf{z}_{k-4}\Big\}.$
$B_k^1$, as in (2.151) and $C_k^1$, $D_k^1$, $F_k^1$, $G_k^1$, and $H_k^1$, as in (2.132),
$\mathbf{z}_k = (B_k' A^2 + C_k' A + D_k')\mathbf{r}_{k-3} + (A^3 + F_k' A^2 + G_k' A + H_k')\mathbf{z}_{k-3}.$
$k = k + 1,$
**EndWhile**
Obtain the approximate solution as well as the residual norm;
$\text{sol}_{last} = \mathbf{x}_k,$
$\text{norm}_{last} = \|\mathbf{r}_k\|.$
**Stop**.

### 2.4.4 Lanczos-type Algorithm Based on $A_{25}/B_{19}$

From relation $A_{25}$ of subsection 2.2.6, Eq (2.70), after replacing $x$ by $A$. Since $\mathbf{r}_k = P_k(A)\mathbf{r}_0$, we have

$$\mathbf{r}_k(x) = \mathbf{r}_{k-3} + (B_k A^2 + C_k A)\mathbf{r}_{k-3} + (E_k A^3 + F_k A^2 + G_k A)\mathbf{z}_{k-3}. \tag{2.152}$$

Using $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, we get

$$\mathbf{x}_k = \mathbf{x}_{k-3} - (B_k A + C_k I)\mathbf{r}_{k-3} - (E_k A^2 + F_k A + G_k)\mathbf{z}_{k-3}. \tag{2.153}$$

Eqs (2.152) and (2.153) define a Lanczos-type algorithm. Now, we have to find the expressions of the coefficients $B_k$, $C_k$, $E_k$, $F_k$, and $G_k$ appearing in them, have been derived in subsection 2.2.6.

The rest of the coefficient can be written explicitly as follow:

$a_{11} = (\mathbf{y}_{k-3}, \mathbf{r}_{k-3}),\ \ a_{13} = (\mathbf{y}_{k-2}, \mathbf{z}_{k-3}),$

$a_{21} = (\mathbf{y}_{k-2}, \mathbf{r}_{k-3}),\ \ a_{22} = a_{11},\ \ a_{23} = (\mathbf{y}_{k-1}, \mathbf{z}_{k-3}),\ \ a_{24} = a_{13},$

$a_{31} = (\mathbf{y}_{k-1}, \mathbf{r}_{k-3}),\ \ a_{32} = a_{21},\ \ a_{33} = (\mathbf{y}_{k}, \mathbf{z}_{k-3}),\ \ a_{34} = a_{23},\ \ a_{35} = a_{24},$

$a_{41} = (\mathbf{y}_{k}, \mathbf{r}_{k-3}),\ \ a_{42} = a_{31},\ \ a_{43} = (\mathbf{y}_{k+1}, \mathbf{z}_{k-3}),\ \ a_{44} = a_{33},\ \ a_{45} = a_{34},$

$a_{51} = (\mathbf{y}_{k+1}, \mathbf{r}_{k-3}),\ \ a_{52} = a_{41},\ \ a_{53} = (\mathbf{y}_{k+2}, \mathbf{z}_{k-3}),\ \ a_{54} = a_{43},\ \ a_{55} = a_{44},$

Using these relations we get

$b_3 = -(\mathbf{y}_{k-3}, \mathbf{r}_{k-3}) = -a_{11},$

$b_4 = -(\mathbf{y}_{k-2}, \mathbf{r}_{k-3}) = -a_{21},$

$b_5 = -(\mathbf{y}_{k-1}, \mathbf{r}_{k-3}) = -a_{31},$

Since all previous formulae are valid for $k \geq 3$, therefor we need $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{x}_1$, and $\mathbf{x}_2$, which are necessary to evaluate (2.152) and (2.153) recursively, which are given as in Eqs (2.138) and (2.139).

From relation $B_{19}$ of subsection 2.4.2, we have

$$\mathbf{z}_k = (C_k^1 A^2 + D_k^1 A + E_k^1)\mathbf{Z}_{k-4} + (A^3 + G_k^1 A^2 + H_k^1 A + I_k^1)\mathbf{z}_{k-3}. \tag{2.154}$$

Note that the coefficients of (2.154) are already derived in subsection 2.4.2. We finally have the following algorithm after gathering together all these formulae.

**Algorithm 4** Lanczos-type Algorithm based on relations $A_{25}/B_{19}$

**Input:** $A$ an $n \times n$ matrix, $\mathbf{b}$ an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $||\mathbf{r}_k||$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, such that $\mathbf{y} \neq 0$ and the tolerance $\varepsilon$ to $1.0E - 13$.
Set $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$; $\mathbf{y}_0 = \mathbf{y}$; $\mathbf{z}_0 = \mathbf{r}_0$.
**Compute:**
$c_0, c_1, c_2, c_3, c_4$ and $c_5$; as in (1.23b).
$\mathbf{r}_1$, $\mathbf{x}_1$, $\mathbf{r}_2$, $\mathbf{x}_2$, $\mathbf{r}_3$ and $\mathbf{x}_3$ as in (2.138), (2.139) and (2.140).
$\mathbf{z}_1$, $\mathbf{z}_2$, and $\mathbf{z}_3$, as in (2.146), (2.147) and (2.148).
$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4$ with $\mathbf{y}_k = A^T\mathbf{y}_{k-1}$.
$k = 3$,
**While** $||\mathbf{r}_k|| > \varepsilon$ **do**
$\mathbf{y}_{k+2} = A^T\mathbf{y}_{k+1}$.
$B_k$, $C_k$, $E_k$, $F_k$, and $G_k$, as in (2.69).
$\mathbf{r}_k = \mathbf{r}_{k-3} + (B_k A^2 + C_k A)\mathbf{r}_{k-3} + (E_k A^3 + F_k A^2 + G_k A)\mathbf{z}_{k-3}$,
$\mathbf{x}_k = \mathbf{x}_{k-3} - (B_k A + C_k)\mathbf{r}_{k-3} - (E_k A^2 + F_k A + G_k)\mathbf{z}_{k-3}$.
$C_k^1$, as in (2.145);
$D_k^1$, $E_k^1$, $G_k^1$, $H_k^1$ and $I_k^1$, as in (2.119),
$\mathbf{z}_k = (C_k' A^2 + D_k' A + E_k')\mathbf{z}_{k-4} + (A^3 + G_k' A^2 + H_k' A + I_k')\mathbf{z}_{k-3}$.
$k = k + 1$,
**EndWhile**
Obtain the approximate solution as well as the residual norm;
$\mathrm{sol}_{last} = \mathbf{x}_k$,
$\mathrm{norm}_{last} = ||\mathbf{r}_k||$.
**Stop**.

### 2.4.5 Lanczos-type Algorithm Based on $A_{25}/B_{21}$

From equations (2.152) and (2.153) of subsection 2.4.4, we have

$$\begin{cases} \mathbf{r}_k = \mathbf{r}_{k-3} + (B_k A^2 + C_k A)\mathbf{r}_{k-3} + (E_k A^3 + F_k A^2 + G_k A)\mathbf{z}_{k-3}., \\ \mathbf{x}_k = \mathbf{x}_{k-3} - (B_k A + C_k I)\mathbf{r}_{k-3} - (E_k A^2 + F_k A + G_k)\mathbf{z}_{k-3}. \end{cases} \tag{2.155}$$

With all coefficients involved have been derived in subsection 2.4.4.

The equation (2.150) of subsection 2.4.3, we have

$$\mathbf{z}_k = (B_k A^2 \mathbf{r}_{k-3} + C_k A \mathbf{r}_{k-3} + D_k \mathbf{r}_{k-3} + A^3 \mathbf{z}_{k-3} + F_k A^2 \mathbf{z}_{k-3} + G_k A \mathbf{z}_{k-3} + H_k \mathbf{z}_{k-3}),$$

with all coefficients involved already derived in subsection 2.4.3. We finally have the following algorithm after gathering together all these formulae.

---

**Algorithm 5** Lanczos-type Algorithm based on relations $A_{25}/B_{21}$

---

**Input:** $A$ an $n \times n$ matrix, **b** an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $||\mathbf{r}_k||$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, such that $\mathbf{y} \neq 0$ and the tolerance $\varepsilon$ to $1.0E-13$.
Set $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$; $\mathbf{y}_0 = \mathbf{y}$; $\mathbf{z}_0 = \mathbf{r}_0$.
**Compute:**
$c_0, c_1, c_2, c_3, c_4$ and $c_5$ as in (1.23b),
$\mathbf{r}_1$, $\mathbf{x}_1$, $\mathbf{r}_2$, $\mathbf{x}_2$, $\mathbf{r}_3$ and $\mathbf{x}_3$ as in (2.138), (2.139) and (2.140),
$\mathbf{z}_1$, $\mathbf{z}_2$, and $\mathbf{z}_3$, as in (2.146), (2.147) and (2.148),
$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4$ with $\mathbf{y}_k = A^T \mathbf{y}_{k-1}$.
$k = 3$.
**While** $||\mathbf{r}_k|| > \varepsilon$ **do**
$\mathbf{y}_{k+2} = A^T \mathbf{y}_{k+1}$,
$B_k$, $C_k$, $E_k$, $F_k$, and $G_k$ as in (2.69),
$\mathbf{r}_k = \mathbf{r}_{k-3} + (B_k A^2 + C_k A)\mathbf{r}_{k-3} + (E_k A^3 + F_k A^2 + G_k A)\mathbf{z}_{k-3}$,
$\mathbf{x}_k = \mathbf{x}_{k-3} - (B_k A + C_k)\mathbf{r}_{k-3} - (E_k A^2 + F_k A + G_k)\mathbf{z}_{k-3}$.
$B_k^1$, as in (2.151);
$C_k^1$, $D_k^1$, $F_k^1$, $G_k^1$, and $H_k^1$, as in (2.132),
$\mathbf{z}_k = (B_k' A^2 + C_k' A + D_k')\mathbf{r}_{k-3} + (A^3 + F_k' A^2 + G_k' A + H_k')\mathbf{z}_{k-3}$.
$k = k + 1$,
**EndWhile**
Obtain the approximate solution as well as the residual norm;
$\text{sol}_{last} = \mathbf{x}_k$,
$\text{norm}_{last} = ||\mathbf{r}_k||$.
**Stop**.

---

### 2.4.6 Lanczos-type Algorithm Based on $A_{28}/B_{19}$

From relation $A_{28}$ of subsection 2.2.9, the equation (2.96), after replacing $x$ by $A$. Since $\mathbf{r}_k = P_k(A)\mathbf{r}_0$, we have

$$\mathbf{r}_k(x) = \mathbf{r}_{k-2} + A_k\{A^2 \mathbf{r}_{k-2} + B_k A \mathbf{r}_{k-2} + E_k A^2 \mathbf{z}_{k-3} + F_k A \mathbf{z}_{k-3}\}. \tag{2.156}$$

Using $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, we get

$$\mathbf{x}_k = \mathbf{x}_{k-2} - A_k\Big\{A\mathbf{r}_{k-2} + B_k\mathbf{r}_{k-2} + E_kA\mathbf{z}_{k-3} + F_k\mathbf{z}_{k-3}\Big\}. \tag{2.157}$$

Equations (2.156) and (2.157) define a Lanczos-type algorithm. Now, we have to find the expressions of the coefficients $A_k$, $B_k$, $C_k$, $E_k$ and $F_k$, appearing in them, have been derived in subsection (2.2.9). Therefore, we can write using equation (2.136) we get

$$E_k = -\frac{(\mathbf{y}_{k-2}, \mathbf{r}_{k-2})}{(\mathbf{y}_{k-2}, \mathbf{z}_{k-3})}. \tag{2.158}$$

The rest of the coefficient can be written explicitly as follow:

$a_{11} = (\mathbf{y}_{k-2}, \mathbf{r}_{k-2}), \quad a_{13} = (\mathbf{y}_{k-2}, \mathbf{z}_{k-3}),$

$a_{21} = (\mathbf{y}_{k-1}, \mathbf{r}_{k-2}), \quad a_{22} = a_{11}, \quad a_{23} = (\mathbf{y}_{k-1}, \mathbf{z}_{k-3}),$

$a_{31} = (\mathbf{y}_k, \mathbf{r}_{k-2}), \; a_{32} = a_{21}, \quad a_{33} = (\mathbf{y}_k, \mathbf{Z}_{k-3})$

Using these relations we get

$b_1 = -a_{21} - E_k a_{23},$

$b_2 = -a_{31} - E_k a_{33},$

$b_3 = -s - tE_k,$ where $s = (\mathbf{y}_{k+1}, \mathbf{r}_{k-2})$ and $t = (\mathbf{y}_{k+1}, \mathbf{z}_{k-3})$.

Equations (2.156) and (2.157) are valid for $k \geq 3$. We need $\mathbf{r}_1$, $\mathbf{x}_1$, $\mathbf{r}_2$ and $\mathbf{x}_2$, which can be evaluated by equations (2.138) and (2.139).

Eq (2.144), from relation $B_{19}$ of subsection 2.4.2, we have

$$\mathbf{z}_k = (C_k^1A^2 + D_k^1A + E_k^1)\mathbf{Z}_{k-4} + (A^3 + G_k^1A^2 + H_k^1A + I_k^1)\mathbf{z}_{k-3}. \tag{2.159}$$

Now, we have to find the expressions of the coefficients $C_k^1$, $D_k^1$, $E_k^1$, $G_k^1$, $H_k^1$ and $I_k^1$ appearing in them, have already been derived in subsection 2.4.2. We finally have the following algorithm after gathering together all these formulae.

**Algorithm 6** Lanczos-type Algorithm based on relations $A_{28}/B_{19}$

**Input:** $A$ an $n \times n$ matrix, **b** an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $||\mathbf{r}_k||$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, such that $\mathbf{y} \neq 0$ and the tolerance $\varepsilon$ to $1.0E-13$.
Set $\mathbf{r}_0 = b - A\mathbf{x}_0$; $\mathbf{y}_0 = \mathbf{y}$; $\mathbf{z}_0 = \mathbf{r}_0$;
**Compute:**
$c_0, c_1, c_2, c_3$, $c_4$ and $c_5$; as in (1.23b),
$\mathbf{r}_1$, $\mathbf{x}_1$, $\mathbf{r}_2$, $\mathbf{x}_2$, $\mathbf{r}_3$ and $\mathbf{x}_3$ as in (2.138), (2.139) and (2.140),
$\mathbf{z}_1$, $\mathbf{z}_2$, and $\mathbf{z}_3$, as in (2.146), (2.147) and (2.148),
$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4$ with $\mathbf{y}_k = A^T\mathbf{y}_{k-1}$.
$k = 3$.

**While** $||\mathbf{r}_k|| > \varepsilon$ **do**
$\mathbf{y}_{k+2} = A^T\mathbf{y}_{k+1}$,
$A_k$, $E_k$, $B_k$, $C_k$, and $F_k$, as in (2.89), (2.158) and (2.95) respectively,
$\mathbf{r}_k = \mathbf{r}_{k-2} + A_k\{A^2\mathbf{r}_{k-2} + B_kA\mathbf{r}_{k-2} + E_kA^2\mathbf{z}_{k-3} + F_kA\mathbf{z}_{k-3}\}$,
$\mathbf{x}_k = \mathbf{x}_{k-2} - A_k\{A\mathbf{r}_{k-2} + B_k\mathbf{r}_{k-2} + E_kA\mathbf{z}_{k-3} + F_k\mathbf{z}_{k-3}\}$.
$C_k^1$, as in (2.145);
$D_k^1$, $E_k^1$, $G_k^1$, $H_k^1$, and $I_k^1$, as in (2.119),
$\mathbf{z}_k = (C_k'A^2 + D_k'A + E_k')\mathbf{z}_{k-4} + (A^3 + G_k'A^2 + H_k'A + I_k')\mathbf{z}_{k-3}$.
$k = k + 1$.
**EndWhile**
Obtain the approximate solution as well as the residual norm;
$\text{sol}_{last} = \mathbf{x}_k$,
$\text{norm}_{last} = ||\mathbf{r}_k||$.
**Stop**.

### 2.4.7 Lanczos-type Algorithm Based on $A_{28}/B_{21}$

From equations (2.156) and (2.157) of subsection 2.4.6, we have

$$\begin{cases} \mathbf{r}_k = \mathbf{r}_{k-2} + A_k\{A^2\mathbf{r}_{k-2} + B_kA\mathbf{r}_{k-2} + E_kA^2z_{k-3} + F_kA\mathbf{z}_{k-3}\}, \\ \mathbf{x}_k = \mathbf{x}_{k-2} - A_k\Big\{A\mathbf{r}_{k-2} + B_k\mathbf{r}_{k-2} + E_kA\mathbf{z}_{k-3} + F_k\mathbf{z}_{k-3}\Big\}. \end{cases} \tag{2.160}$$

with all coefficients involved being already derived in subsection 2.4.6.

From Eq 2.150, of subsection 2.4.3, we have

$$\mathbf{z}_k = (B_k^1A^2\mathbf{r}_{k-3} + C_k^1A\mathbf{r}_{k-3} + D_k^1\mathbf{r}_{k-3} + A^3\mathbf{z}_{k-3} + F_k^1A^2\mathbf{z}_{k-3} + G_k^1A\mathbf{z}_{k-3} + H_k^1\mathbf{z}_{k-3}), \tag{2.161}$$

with all coefficients involved having been derived in subsection 2.4.3. We finally have the following algorithm after gathering together all these formulae.

**Algorithm 7** Lanczos-type Algorithm based on relations $A_{28}/B_{21}$

**Input:** $A$ an $n \times n$ matrix, $\mathbf{b}$ an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $||\mathbf{r}_k||$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, such that $\mathbf{y} \neq 0$ and the tolerance $\varepsilon$ to $1.0E-13$.
Set $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$; $\mathbf{y}_0 = \mathbf{y}$; $\mathbf{z}_0 = \mathbf{r}_0$.
**Compute:**
$c_0, c_1, c_2, c_3, c_4$ and $c_5$; as in (1.23b),
$\mathbf{r}_1$, $\mathbf{x}_1$, $\mathbf{r}_2$, $\mathbf{x}_2$, $\mathbf{r}_3$ and $\mathbf{x}_3$ as in (2.138), (2.139) and (2.140),
$\mathbf{z}_1$, $\mathbf{z}_2$, and $\mathbf{z}_3$, as in (2.146), (2.147) and (2.148),
$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4$ with $\mathbf{y}_k = A^T\mathbf{y}_{k-1}$.
$k = 3$.
**While** $||\mathbf{r}_k|| > \varepsilon$ **do**
$\mathbf{y}_{k+2} = A^T\mathbf{y}_{k+1}$,
$A_k$, $E_k$, $B_k$, $C_k$, and $F_k$, as in (2.89), (2.158) and (2.95) respectively,
$\mathbf{r}_k = \mathbf{r}_{k-2} + A_k\{A^2\mathbf{r}_{k-2} + B_kA\mathbf{r}_{k-2} + E_kA^2\mathbf{z}_{k-3} + F_kA\mathbf{z}_{k-3}\}$,
$\mathbf{x}_k = \mathbf{x}_{k-2} - A_k\{A\mathbf{r}_{k-2} + B_k\mathbf{r}_{k-2} + E_kA\mathbf{z}_{k-3} + F_k\mathbf{z}_{k-3}\}$.
$B_k^1$, as in (2.151);
$C_k^1$, $D_k^1$, $F_k^1$, $G_k^1$, and $H_k^1$, as in (2.132)
$\mathbf{z}_k = (B_k'A^2 + C_k'A + D_k')\mathbf{r}_{k-3} + (A^3 + F_k'A^2 + G_k'A + H_k')\mathbf{z}_{k-3}$.
$k = k + 1$.
**EndWhile**
Obtain the approximate solution as well as the residual norm.
$\text{sol}_{last} = \mathbf{x}_k$,
$\text{norm}_{last} = ||\mathbf{r}_k||$.
**Stop**.

## 2.5 Numerical results of $A_{20}$, $A_{22}/B_{19}$, $A_{22}/B_{21}$ and $A_{28}/B_{19}$

We have solved different small size problems [4, 33]. These algorithms are coded out in Matlab R2014b and run on a PC under the Microsoft Windows 7 Enterprise, with 16.00GB RAM, and processor Intel(R) Core(TM) i5-3570 CPU 3.40GHz. Experimental results obtained on the test problem $Ax = b$ with $A$ refer to the Baheux-typ problems [4] as below are recorded in the following table. The stoping criteria is the norm of residual

$||r_k|| = eps = 1.0E - 13$

$$A = \begin{pmatrix} B & -I & \cdots & \cdots & 0 \\ -I & B & -I & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & -I & B & -I \\ 0 & \cdots & \cdots & -I & B \end{pmatrix}, \quad \text{with} \quad B = \begin{pmatrix} 4 & \alpha & \cdots & \cdots & 0 \\ \beta & 4 & \alpha & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \beta & 4 & \alpha \\ 0 & \cdots & \cdots & \beta & 4 \end{pmatrix}$$

and $\alpha = -1 + \delta$, $\beta = -1 - \delta$. The parameter $\delta$ takes the value 0 and thus the matrix $A$ is symmetric and the problem is easy to solve because the region is a regular mesh. While for all other values of $\delta$ the matrix $A$ becomes non-symmetric and the problem is relatively harder to solve as the region is not regular mesh. The right hand side $b$ is taken to be $b = AX$, where $X = (1, 1, ...1)^T$, is the solution of the system. The dimension of B is 10. The computational results obtained with algorithms $A_{20}$, $A_{22}/B_{21}$, $A_{25}/B_{19}$ and $A_{28}/B_{19}$ are recorded in Table 2.1.

**Table 2.1:** *Results of $A_{20}$, $A_{22}/B_{21}$, $A_{25}/B_{19}$ and $A_{28}/B_{19}$, on Baheux-type problems when $\delta = 0$*

| Dim of Prob | $A_{20}$ | | $A_{22}/B_{21}$ | | $A_{25}/B_{19}$ | | $A_{28}/B_{19}$ | |
|---|---|---|---|---|---|---|---|---|
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | sec | $\|\|r_k\|\|$ | sec | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | sec |
| 10 | 1.9828e-14 | 1.2818E-02 | 1.5104E-14 | 4.4420E-03 | 5.0861E-14 | 7.4318E-03 | 3.8274E-14 | 6.29054E-03 |
| 20 | NaN | | 2.5648E-14 | 5.8613E-03 | 4.0278E-14 | 2.1781E-03 | 8.9743E-14 | 7.5220E-04 |
| 50 | NaN | | NaN | | NaN | | NaN | |
| 100 | NaN | | NaN | | NaN | | NaN | |
| 500 | NaN | | NaN | | NaN | | NaN | |
| 1000 | NaN | | NaN | | NaN | | NaN | |

The experimental results which are recorded in the Table 2.1 show that algorithms $A_{22}/B_{21}$, $A_{25}/B_{21}$ and $A_{28}/B_{19}$ solved the problems with up to dimension 20. These algorithms failed for $n \geq 30$. The reason is obvious, it is due to a division by zero that can not be avoided when computing the coefficients of those recurrence relations based on $P_k(x)$ and

$P_k^{(1)}(x)$. Some of the scalar products in the denominator are as small as E-14, which causes the breakdown of these Lanczos-type of algorithms and the algorithms have generally to be stopped. Equivalently, in the recursive computation of FOPs, a breakdown can be caused by the non-existence of some coefficients of the FOPs involved in the recurrence relations. Restarting is used to avoid the problem. This strategy either stops the Lancozs-type algorithm pre-emptively and restarts it with some iterate or waits until the breakdown occurs and then restarts from the last iterate found. Various Krylov subspaces are considered for the algorithm to start working. The existing algorithms $A_4$, and $A_{12}$ are considered the most robust Lanczos-type algorithms according to [4, 33]. Therefore, we have compared our new algorithms $A_{20}$ with these on the standard problems considered in [3, 33]. These breakdowns are mainly of two types:

1. Since all algorithms of this type are based on recurrence relationships between FOPs $P_k(x)$, these polynomials involve the computation of some scalar products appearing as denominators and numerators of the coefficients of the recursive relationships, with some of the denominators becoming smaller than $1.000E - 14$ which causes a breakdown in these algorithms and the algorithms have to be stopped.

2. The breakdown is due to the non-existence of some polynomials $P_k(x)$.

## 2.6 Restarting Lanczos-type Algorithm based on relation $A_{20}$

The solution is obtained via restarting algorithm $A_{20}$ as given in Algorithm 8. Utilizing regular intervals, the algorithm is restarted using the current iterate. The restarting procedure can be described follows.

**Algorithm 8** Restarting Lanczos-type Algorithm based on relation $A_{20}$

Run **Algorithm 1** for a fixed number of iterations $k$ or until it halts and obtain the approximate solution $sol_{last} = \mathbf{x}_k$ as well as the residual norm $norm_{last} = ||\mathbf{r}_k||$.
**While** $||\mathbf{r}_k|| > \varepsilon$ **do**
initialize it with the current iterate of the algorithm run,
$\mathbf{x} = sol_{last}$,
$\mathbf{y} = \mathbf{b} - A\mathbf{x}$.
Run **Algorithm 1** for a fixed number of iterations $k$
**EndWhile**
Obtain the optimal solution as well as the optimal residual norm as follows
$sol_{optimal} = \mathbf{x}_k$
$norm_{optimal} = ||\mathbf{r}_k||$.
**Stop**.

### 2.6.1 Numerical results

The results obtained with Algorithm 8, restarting algorithm $A_{20}$ on Baheux-type problems of different dimensions, for different values of $\delta = 0$ [3, 4] are presented in Table 2.2.

**Table 2.2:** *Results of $A_{20}$, $A_4$ and $A_{12}$ on Baheux-type problems when $\delta = 0$*

| Dim of Prob | $A_{20}$ | | | $A_4$ | | | $A_{12}$ | | |
|---|---|---|---|---|---|---|---|---|---|
| $n_1 \times n_2 = n$ | cycles | $\|\|r_k\|\|$ | sec | cycles | $\|\|r_k\|\|$ | sec | cycles | $\|\|r_k\|\|$ | sec |
| 10 | 1 | 1.9828E-14 | 5.5432E-01 | 1 | 3.7525E-14 | 3.6798E-01 | 1 | 2.2493E-14 | 3.8486E-01 |
| 50 | 3 | 6.2988E-14 | 5.9230E-01 | 2 | 2.7427E-14 | 5.0729E-01 | 2 | 9.8576E-15 | 4.6922E-01 |
| 100 | 3 | 3.8627E-14 | 1.4741E+00 | 2 | 4.5148E-14 | 5.7386E-01 | 3 | 6.2923E-14 | 6.4445E-01 |
| 500 | 10 | 9.6832E-14 | 4.8584E+00 | 11 | 9.2011E-14 | 7.7663E-01 | 10 | 8.6145E-14 | 8.2438E-01 |
| 1000 | 11 | 7.8684E-14 | 2.8250E+01 | 11 | 8.3822E-14 | 1.2431E+00 | 10 | 8.2999E-14 | 1.5196E+00 |
| 2000 | 11 | 7.5277E-14 | 2.2839E+02 | 10 | 8.5165E-14 | 2.0823E+00 | 12 | 9.2854E-14 | 3.4794E+00 |
| 3000 | 11 | 9.9051E-14 | 5.5966E+02 | 10 | 8.8804E-14 | 3.5308E+00 | 11 | 8.5873E-14 | 6.0836E+00 |
| 4000 | 11 | 8.9856E-14 | 1.3869E+03 | 10 | 9.3931E-14 | 5.5072E+00 | 10 | 8.2973E-14 | 1.2737E+01 |
| 5000 | 11 | 9.1068E-14 | 2.3177E+03 | 10 | 9.6259E-14 | 7.7054E+00 | 13 | 8.8102E-14 | 8.5873E+01 |

The Lanczos algorithm based on $A_{20}$ involves higher degree FOPs, which means that many coefficients have to be estimated compared to $A_4$ and $A_{12}$ for instance in $A_{20}$, $A_{12}$ and $A_4$, 7 , 5 and 3 are the number of coefficients respectively. This means error accumulation, loss of orthogonality and ultimately breakdown are likely to occur. For this reason only low dimensional problems can be solved without a remedial approach.

## 2.7 Summary

This chapter looked at new recurrence relations between FOP's in a systematic fashion where some of the relations might lead to new Lanczos-type algorithms. The expression of their coefficients have also been derived. The recurrence relations investigated here were not studied before. It was observed that relations $A_{21}$, $A_{23}$, $A_{24}$, $A_{26}$, $A_{27}$ $B_{17}$, $B_{18}$, and $B_{20}$ do not exist, while, relations $A_{23}$, $A_{24}$, and $A_{27}$ do exist but could not be used for deriving Lanczos-type algorithms. Relations $A_{20}$, $A_{22}$, $A_{25}$, $A_{28}$, $B_{19}$, and $B_{21}$ exist and were found suitable for the implementation of new Lanczos-type algorithms. Relation $A_{20}$ alone led to a new Lanczos-type algorithm while the other relations can make new Lanczos-type algorithms when combined in $A_i/B_j$ manner. Possible combinations are:

$A_{22}/B_{19}$, $A_{22}/B_{21}$,

$A_{25}/B_{19}$, $A_{25}/B_{21}$,

$A_{28}/B_{19}$, $A_{28}/B_{21}$.

All the algorithms mentioned above need $P_k(x)$ for the derivation of $r_k$ and $P_k^{(1)}(x)$ for $z_k$ except $A_{20}$. Algorithms $A_{20}$, $A_4$ and $A_{12}$ are tested on some problems of small size. The results of Algorithm $A_{20}$ have been compared on problems of various sizes with algorithms $A_4$ and $A_{12}$ Lanczos-type algorithms.

# Chapter 3

# New Recurrence Relations for the Different Choice of Unit Polynomials $U_i(x)$

## 3.1 Introduction

In this chapter we derive new recurrence relationships between the adjacent orthogonal polynomials for the different choices of unit polynomial $U_i(x) = P_i(x)$ and $U_i(x) = P_i^{(1)}(x)$, that can be used in the derivation of new Lanczos-type algorithms [33].

## 3.2 Formula $A_i$ when $U_i(x) = P_i(x)$

Consider the formulae of type $A_i$ for the choice of $U_i(x) = P_i(x)$, which have not been considered before [33]. These formulae will be used in combination with formulae $B_j$ to derive new Lanczos-type algorithms.

### 3.2.1 Formula $A_{13new}$

Consider the following recurrence relationship for $k \geq 3$,

$$P_k(x) = A_k\Big\{(x^2 + B_k x + C_k)P_{k-2} + (D_k x^3 + E_k x^2 + F_k x + G_k)P^{(1)}_{k-3}\Big\}, \tag{3.1}$$

where $P_k(x)$, $P_{k-2}(x)$ and $P^{(1)}_{k-3}(x)$ are polynomials of degree $k$, $k-2$ and $k-3$ respectively. The constant coefficients $A_k$, $B_k$, $C_k$, $D_k$, $E_k$,$F_k$ and $G_k$ are determined by $P_k(0) = 1$ and imposing the orthogonality condition (2.1) with respect to the linear function $c$. Since $P_k(0) = 1$, $\forall\, k$, then for $x = 0$, equation (3.1) becomes

$$A_k\{C_k + G_k P^{(1)}_{k-3}\} = 1. \tag{3.2}$$

After multiplying equation (3.1) by $U_i$ a polynomial of exact degree $i$ and applying linear functional $c$ on both sides it becomes

$$c(U_iP_k) = A_k\Big\{c(x^2U_iP_{k-2}) + B_kc(xU_iP_{k-2}) + C_kc(U_iP_{k-2}) + D_kc(x^3U_iP^{(1)}_{k-3}) + E_kc(x^2U_iP^{(1)}_{k-3}) + F_kc(xU_iP^{(1)}_{k-3}) + G_kc(U_iP^{(1)}_{k-3})\Big\}. \tag{3.3}$$

Consequently, by applying (2.1), we have the relation for $i = 0, 1, ..., k-1$.

$$c(x^2U_iP_{k-2}) + B_kc(xU_iP_{k-2}) + C_kc(U_iP_{k-2}) + D_kc(x^3U_iP^{(1)}_{k-3}) + E_kc(x^2U_iP^{(1)}_{k-3}) + F_kc(xU_iP^{(1)}_{k-3}) + G_kc(U_iP^{(1)}_{k-3}) = 0.$$

$$c(x^2U_iP_{k-2}) + B_kc(xU_iP_{k-2}) + C_kc(U_iP_{k-2}) + D_kc^{(1)}(x^2U_iP^{(1)}_{k-3}) + E_kc^{(1)}(xU_iP^{(1)}_{k-3}) + F_kc^{(1)}(U_iP^{(1)}_{k-3}) + G_kc(U_iP^{(1)}_{k-3}) = 0. \tag{3.4}$$

For $i = 0$, equation (3.4) becomes $G_kc(U_0P^{(1)}_{k-3}) = 0$, since

$$c(U_0P^{(1)}_{k-3}) \neq 0 \;\Rightarrow\quad G_k = 0.$$

Therefore, from (3.2) we have

$$A_k = \frac{1}{C_k}.$$

The orthogonality condition is always true for $i = 0, 1, 2, ......, k-6$.

For $i = k-5$, equation (3.4) gives

$$D_k c^{(1)}(x^2 U_{k-5} P^{(1)}_{k-3}) = 0.$$

$$\Rightarrow \quad c^{(1)}(x^2 U_{k-5} P^{(1)}_{k-3}) \neq 0, \quad D_k = 0. \tag{3.5}$$

For $i = k-4$, equation (3.4) gives

$$c(x^2 U_{k-4} P_{k-2}) + E_k c^{(1)}(x U_{k-4} P^{(1)}_{k-3}) = 0,$$

$$E_k = -\frac{c(x^2 U_{k-4} P_{k-2})}{c(x^2 U_{k-4} P^{(1)}_{k-3})}. \tag{3.6}$$

For $i = k-3$, equation (3.4) gives

$$B_k c(x U_{k-3} P_{k-2}) + F_k c^{(1)}(U_{k-3} P^{(1)}_{k-3}) = -c(x^2 U_{k-3} P_{k-2}) - E_k c^{(1)}(x U_{k-3} P^{(1)}_{k-3}) \tag{3.7}$$

For $i = k-2$, equation (3.4) gives

$$B_k c(x U_{k-2} P_{k-2}) + C_k c(U_{k-2} P_{k-2}) + F_k c^{(1)}(U_{k-2} P^{(1)}_{k-3}) = -c(x^2 U_{k-2} P_{k-2}) - E_k c^{(1)}(x U_{k-2} P^{(1)}_{k-3}). \tag{3.8}$$

For $i = k-1$, and equation (3.4) gives

$$B_k c(x U_{k-1} P_{k-2}) + C_k c(U_{k-1} P_{k-2}) + F_k c^{(1)}(U_{k-1} P^{(1)}_{k-3}) = -c(x^2 U_{k-1} P_{k-2}) - E_k c^{(1)}(x U_{k-1} P^{(1)}_{k-3}). \tag{3.9}$$

Equations (3.7), (3.8) and (3.9) can be written as

$$\begin{cases} a_{11} B_k + a_{13} F_k = b_1, \\ a_{21} B_k + a_{22} C_k + a_{23} F_k = b_2, \\ a_{31} B_k + a_{32} C_k + a_{33} F_k = b_3. \end{cases} \tag{3.10}$$

Where $a_{11}$, $a_{13}$, $a_{21}$, $a_{22}$, $a_{23}$, $a_{31}$, $a_{32}$, $a_{33}$, are the coefficients of $B_k$, $C_k$, and $F_k$. Suppose $b_1$, $b_2$, and $b_3$ are the corresponding right hand side terms of these equations.

$$\begin{cases} b_1 = -c(x^2 U_{k-3} P_{k-2}) - E_k c(x^2 U_{k-3} P^{(1)}_{k-3}), \\ b_2 = -c(x^2 U_{k-2} P_{k-2}) - E_k c(x^2 U_{k-2} P^{(1)}_{k-3}), \\ b_3 = -c(x^2 U_{k-1} P_{k-2}) - E_k c(x^2 U_{k-1} P^{(1)}_{k-3}). \end{cases} \tag{3.11}$$

If $\Delta_k$ represents the determinant of the coefficients matrix of (3.10) then we have

$$\Delta_k = a_{11}(a_{22}a_{33} - a_{23}a_{32}) + a_{13}(a_{21}a_{32} - a_{31}a_{22}). \tag{3.12}$$

If $\Delta_k \neq 0$, then

$$\begin{cases} B_k = \frac{1}{\Delta_k}\{b_1(a_{22}a_{33} - a_{23}a_{32}) + a_{13}(b_2a_{32} - b_3a_{22})\}, \\ C_k = \frac{b_2 - a_{21}B_k - F_ka_{23}}{a_{22}}, \\ F_k = \frac{b_1 - a_{11}B_k}{a_{13}}, \\ A_k = \frac{1}{C_k}. \end{cases} \tag{3.13}$$

Since, $D_k = G_k = 0$, relation $A_{13new}$ becomes

$$P_k(x) = A_k\Big\{(x^2 + B_kx + C_k)P_{k-2}(x) + (E_kx^2 + F_kx)P^{(1)}_{k-3}(x)\Big\}. \tag{3.14}$$

Therefore $A_{13new}$ can lead to a Lanczps-type algorithm.

### 3.2.2 Formula $A_{16new}$

Consider the following recurrence relationship for $k \geq 2$,

$$P_k(x) = (A_kx^2 + B_kx + C_k)P_{k-2} + (D_kx^2 + E_kx + F_k)P^{(1)}_{k-2}, \tag{3.15}$$

where $P_k(x)$, $P_{k-2}(x)$ and $P^{(1)}_{k-2}(x)$ are polynomials of degree $k$, $k-2$ and $k-2$ respectively. The constant coefficients $A_k$, $B_k$, $C_k$, $D_k$, $E_k$,and $F_k$ are determined by $P_k(0) = 1$ and imposing the orthogonality condition (2.1) with respect to the linear function $c$. Since $P_k(0) = 1$, $\forall\ k$, then for $x = 0$, equation (3.15) becomes

$$C_k + F_kP^{(1)}_{k-2} = 1. \tag{3.16}$$

After multiplying by $U_i$ a polynomial of exact degree $i$ and applying linear functional $c$ on both sides it becomes

$$\begin{aligned} c(U_iP_k) = A_kc(x^2U_iP_{k-2}) + B_kc(xU_iP_{k-2}) + C_kc(U_iP_{k-2}) + D_kc(x^2U_iP^{(1)}_{k-2}) \\ +E_kc(xU_iP^{(1)}_{k-2}) + F_kc(U_iP^{(1)}_{k-2}). \end{aligned} \tag{3.17}$$

Consequently, by applying (2.1), we have the relation for $i = 0, 1, ..., k-1$,

$$A_k c(x^2 U_i P_{k-2}) + B_k c(xU_i P_{k-2}) + C_k c(U_i P_{k-2}) + D_k c(x^2 U_i P^{(1)}_{k-2}) + E_k c(xU_i P^{(1)}_{k-2})$$
$$+F_k c(U_i P^{(1)}_{k-2}) = 0,$$

$$A_k c(x^2 U_i P_{k-2}) + B_k c(xU_i P_{k-2}) + C_k c(U_i P_{k-2}) + D_k c^{(1)}(xU_i P^{(1)}_{k-2}) + E_k c^{(1)}(U_i P^{(1)}_{k-2})$$
$$+F_k c(U_i P^{(1)}_{k-2}) = 0. \tag{3.18}$$

For $i = 0$, Eq (3.18) becomes $F_k c(U_0 P^{(1)}_{k-2}) = 0$. Since $c(U_0 P^{(1)}_{k-2}) \neq 0 \Rightarrow F_k = 0$, therefore, from (3.16) we have

$$C_k = 1.$$

The orthogonality condition is always true for $i = 0, 1, 2, ......, k-5$.

For $i = k-4$, equation (3.18) gives

$$A_k c(x^2 U_{k-4} P_{k-2}) = 0 \quad \Rightarrow \quad c^{(1)}(x^2 U_{k-4} P^{(1)}_{k-3}) \neq 0, \quad A_k = 0.$$

For $i = k-3$, equation (3.18) gives

$$B_k c(xU_{k-3} P_{k-2}) + D_k c^{(1)}(xU_{k-3} P^{(1)}_{k-2}) = 0. \tag{3.19}$$

For $i = k-2$, equation (3.18) gives

$$B_k c(xU_{k-2} P_{k-2}) + D_k c^{(1)}(xU_{k-2} P^{(1)}_{k-2} + E_k c^{(1)}(U_{k-2} P^{(1)}_{k-2}) = -c(U_{k-2} P_{k-2}). \tag{3.20}$$

For $i = k-1$, equation (3.18) gives

$$B_k c(xU_{k-1} P_{k-2}) + D_k c^{(1)}(xU_{k-1} P^{(1)}_{k-2}) + E_k c^{(1)}(U_{k-1} P^{(1)}_{k-2}) = -c(U_{k-1} P_{k-2}). \tag{3.21}$$

Equations (3.19), (3.20) and (3.21) can be written as

$$\begin{cases} a_{11} B_k + a_{12} D_k = 0, \\ a_{21} B_k + a_{22} D_k + a_{23} E_k = b_2, \\ a_{31} B_k + a_{32} D_k + a_{33} E_k = b_3. \end{cases} \tag{3.22}$$

Where $a_{11}, a_{12}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}, a_{33}$, are the coefficients of $B_k$, $D_k$, and $E_k$. Suppose $b_1$, $b_2$, and $b_3$ are the corresponding right hand side terms of these equations.

$$\begin{cases} b_1 = 0, \\ b_2 = -c(U_{k-2}P_{k-2}), \\ b_3 = -c(U_{k-1}P_{k-2}). \end{cases} \tag{3.23}$$

If $\Delta_k$ represents the determinant of the coefficients matrix of (3.22) then we have

$$\Delta_k = a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{31}a_{23}).$$

If $\Delta_k \neq 0$, then

$$\begin{cases} B_k = \frac{a_{12}(b_3a_{23}-b_2a_{33})}{\Delta_k}, \\ D_k = -\frac{a_{11}B_k}{a_{12}}, \\ E_k = \frac{b_2-a_{21}B_k-D_ka_{22}}{a_{23}}. \end{cases} \tag{3.24}$$

Since, $A_k = F_k = 0$, relation $A_{16new}$ becomes

$$P_k(x) = (B_kx + 1)P_{k-2}(x) + (D_kx^2 + E_kx)P^{(1)}_{k-2}(x). \tag{3.25}$$

Therefore, $A_{16new}$ can lead to a Lanczos-type algorithm.

### 3.2.3 Formula $A_{19new}$

Consider the following recurrence relationship for $k \geq 2$,

$$P_k(x) = (A_kx^2 + B_kx + C_k)P^{(1)}_{k-2} + (D_kx + E_k)P_{k-1}, \tag{3.26}$$

where $P_k(x)$, $P^{(1)}_{k-2}(x)$ and $P_{k-1}(x)$ are polynomials of degree $k$, $k-2$ and $k-1$ respectively. The constant coefficients $A_k$, $B_k$, $C_k$, $D_k$, and $E_k$, are determined by $P_k(0) = 1$ and imposing the orthogonality condition (2.1) with respect to the linear function $c$. Since $P_k(0) = 1$, $\forall\, k$, then for $x = 0$, equation (3.26) becomes

$$C_kP^{(1)}_{k-2} + E_k = 1. \tag{3.27}$$

After multiplying equation (3.26) by $U_i$ a polynomial of exact degree $i$ and applying linear functional $c$ on both sides it becomes

$$c(U_iP_k) = A_kc(x^2U_iP^{(1)}_{k-2}) + B_kc(xU_iP^{(1)}_{k-2}) + C_kc(U_iP^{(1)}_{k-2}) + D_kc(xU_iP_{k-1}) + E_kc(U_iP_{k-1}).$$

Consequently, by applying (2.1), we have the relation for $i = 0, 1, ..., k-1$

$$\begin{aligned} A_kc(x^2U_iP^{(1)}_{k-2}) + B_kc(xU_iP^{(1)}_{k-2}) + C_kc(U_iP^{(1)}_{k-2}) + D_kc(xU_iP_{k-1}) + E_kc(U_iP_{k-1}) = 0, \\ A_kc^{(1)}(xU_iP^{(1)}_{k-2}) + B_kc^{(1)}(U_iP^{(1)}_{k-2}) + C_kc(U_iP^{(1)}_{k-2}) + D_kc(xU_iP_{k-1}) + E_kc(U_iP_{k-1}) = 0. \end{aligned} \tag{3.28}$$

Equation (3.28) is always true $i = 0, 1, 2, ..., k-4$. For $i = 0$, equation (3.28) becomes

$$C_kc(U_0P^{(1)}_{k-2}) = 0, \Rightarrow c(U_0P^{(1)}_{k-2}) \neq 0 \Rightarrow C_k = 0.$$

Therefore, from (3.27) we have $E_k = 1$.

For $i = k-3$, equation (3.28) gives

$$A_kc^{(1)}(xU_{k-3}P^{(1)}_{k-2}) = 0 \Rightarrow c^{(1)}(xU_{k-3}P^{(1)}_{k-3}) \neq 0, \Rightarrow A_k = 0.$$

For $i = k-2$, equation (3.28) gives

$$B_kc^{(1)}(U_{k-2}P^{(1)}_{k-2}) + D_kc(xU_{k-2}P_{k-1}) = 0. \tag{3.29}$$

For $i = k-1$, equation (3.28) gives

$$B_kc^{(1)}(U_{k-1}P^{(1)}_{k-2}) + D_kc(xU_{k-1}P_{k-1}) + E_kc(U_{k-1}P_{k-1}) = 0$$

$\because$ $E_k = 1$, therefore

$$B_kc^{(1)}(U_{k-1}P^{(1)}_{k-2}) + D_kc(xU_{k-1}P_{k-1}) = -c(U_{k-1}P_{k-1}). \tag{3.30}$$

Equations (3.29) and (3.30) can be written as

$$\begin{cases} a_{11}B_k + a_{12}D_k = 0, \\ a_{21}B_k + a_{22}D_k = b_2. \end{cases} \tag{3.31}$$

Where $a_{11}, a_{12}, a_{21}$, and $a_{22}$, are the coefficients of $B_k$, and $D_k$. Suppose $b_2$, is the corresponding right hand side term of these equations.

$$b_2 = -c(U_{k-1}P_{k-1}). \tag{3.32}$$

If $\Delta_k$ represents the determinant of the coefficients matrix of (3.31) then we have

$$\Delta_k = a_{11}a_{22} - a_{12}a_{21}.$$

If $\Delta_k \neq 0$, then

$$\begin{cases} B_k = -\frac{a_{12}b_2}{\Delta_k}, \\ D_k = \frac{a_{11}b_2}{\Delta_k}. \end{cases} \tag{3.33}$$

Since, $A_k = C_k = 0$, relation $A_{19new}$ becomes

$$P_k(x) = B_k x P_{k-2}^{(1)}(x) + (D_k x + I)P_{k-1}(x). \tag{3.34}$$

Therefore, $A_{new19}$ can lead to a Lanczos-type algorithm.

## 3.3 Formula $B_j$ when $U_i(x) = P_i(x)$

Now we consider the formulae of type $B_j$ for the choice of $U_i(x) = P_i(x)$, which have not been considered before [33]. These formulae will be used in combination with formulae $A_i$ to derive new Lanczos-type algorithms.

### 3.3.1 Formula $B_{13new}$

Consider the following recurrence relationship for $k \geq 3$,

$$P_k^{(1)} = (A_k^1 x^3 + B_k^1 x^2 + C_k^1 x + D_k^1)P_{k-3}^{(1)} + (E_k^1 x^2 + F_k^1 x + G_k^1)P_{k-2}^{(1)}, \tag{3.35}$$

where $P_k^{(1)}$, $P_{k-2}^{(1)}$ and $P_{k-3}^{(1)}$ are polynomials of degree $k$, $k-2$ and $k-3$ respectively. The constant coefficients $A_k^1$, $B_k^1$, $C_k^1$, $D_k^1$, $E_k^1$, $F_k^1$ and $G_k^1$ are to be determined by imposing the orthogonality condition (2.2) with respect to the linear function $c^{(1)}$.

Multiplying equation (3.35) by $U_i$ a polynomial of exact degree $i$ and applying linear functional $c^{(1)}$ on both sides it becomes

$$c^{(1)}(U_iP_k^{(1)}) = A_k^1c^{(1)}(x^3U_iP_{k-3}^{(1)}) + B_k^1c^{(1)}(x^2U_iP_{k-3}^{(1)}) + C_k^1c^{(1)}(xU_iP_{k-3}^{(1)}) + D_k^1c^{(1)}(U_iP_{k-3}^{(1)}) + E_k^1c^{(1)}(x^2U_iP_{k-2}^{(1)}) + F_k^1c^{(1)}(xU_iP_{k-2}^{(1)}) + G_k^1c^{(1)}(U_iP_{k-2}^{(1)}).$$

Consequently, by applying (2.2), we have the relation for $i = 0, 1, ..., k-1$.

$$A_k^1c^{(1)}(x^3U_iP_{k-3}^{(1)}) + B_k^1c^{(1)}(x^2U_iP_{k-3}^{(1)}) + C_k^1c^{(1)}(xU_iP_{k-3}^{(1)}) + D_k^1c^{(1)}(U_iP_{k-3}^{(1)}) + E_k^1c^{(1)}(x^2U_iP_{k-2}^{(1)}) + F_k^1c^{(1)}(xU_iP_{k-2}^{(1)}) + G_k^1c^{(1)}(U_iP_{k-2}^{(1)}) = 0 \tag{3.36}$$

The orthogonality condition is always true for $i = 0, 1, 2, ......, k-7$.

For $i = k-6$, equation (3.36) gives

$$A_k^1c^{(1)}(x^3U_{k-6}P_{k-3}^{(1)}) = 0 \quad \Rightarrow \quad c^{(1)}(x^3U_{k-6}P_{k-3}^{(1)}) \neq 0, \quad A_k^1 = 0.$$

For $i = k-5$, equation (3.36) gives

$$B_k^1c^{(1)}(x^2U_{k-5}P_{k-3}^{(1)}) = 0 \quad \Rightarrow \quad c^{(1)}(x^2U_{k-5}P_{k-3}^{(1)}) \neq 0, \quad B_k^1 = 0.$$

For $i = k-4$, equation (3.36) gives

$$C_k^1c^{(1)}(xU_{k-4}P_{k-3}^{(1)}) + E_k^1c^{(1)}(x^{(2)}U_{k-4}P_{k-2}^{(1)}) = 0.$$

Since $P_k^{(1)}$ is a monic polynomial of degree $k$, therefore, $E_k = 1$.

$$C_k^1c^{(1)}(xU_{k-4}P_{k-3}^{(1)}) + c^{(1)}(x^2U_{k-4}P_{k-2}^{(1)}) = 0,$$

$$C_k^1 = -\frac{c(x^3U_{k-4}P_{k-2}^{(1)})}{c(x^2U_{k-4}P_{k-3}^{(1)})}. \tag{3.37}$$

For $i = k-3$, equation (3.36) gives

$$D_k^1c^{(1)}(U_{k-3}P_{k-3}^{(1)}) + F_k^1c^{(1)}(xU_{k-3}P_{k-2}^{(1)}) = -c^{(1)}(x^2U_{k-3}P_{k-2}^{(1)}) - C_k^1c^{(1)}(xU_{k-3}P_{k-3}^{(1)}). \tag{3.38}$$

For $i = k-2$, equation (3.36) gives

$$D_k^1 c^{(1)}(U_{k-2}P_{k-3}^{(1)}) + F_k^1 c^{(1)}(xU_{k-2}P_{k-2}^{(1)}) + G_k^1 c^{(1)}(U_{k-2}P_{k-2}^{(1)}) = -c^{(1)}(x^2U_{k-2}P_{k-2}^{(1)}) - C_k^1 c^{(1)}(xU_{k-2}P_{k-3}^{(1)}). \tag{3.39}$$

For $i = k-1$, and equation (3.36) gives

$$D_k^1 c^{(1)}(U_{k-1}P_{k-3}^{(1)}) + F_k^1 c^{(1)}(xU_{k-1}P_{k-2}^{(1)}) + G_k^1 c^{(1)}(U_{k-1}P_{k-2}^{(1)}) = -c^{(1)}(x^2U_{k-1}P_{k-2}^{(1)}) - C_k^1 c^{(1)}(xU_{k-1}P_{k-3}^{(1)}). \tag{3.40}$$

Equations (3.38), (3.39) and (3.40) can be written as

$$\begin{cases} a'_{11}D_k^1 + a'_{12}F_k^1 = b'_1, \\ a'_{21}D_k^1 + a'_{22}F_k^1 + a'_{23}G_k^1 = b'_2, \\ a'_{31}D_k^1 + a'_{32}F_k^1 + a'_{33}G_k^1 = b'_3. \end{cases} \tag{3.41}$$

Where $a'_{11}, a'_{12}, a'_{21}, a'_{22}, a'_{23}, a'_{31}, a'_{32}, a'_{33}$, are the coefficients of $D_k^1$, $F_k^1$, and $G_k^1$. Suppose $b'_1$, $b'_2$, and $b'_3$ are the corresponding right hand side terms of these equations.

$$\begin{cases} b'_1 = -c^{(1)}(x^2U_{k-3}P_{k-2}^{(1)}) - C_k^1 c^{(1)}(xU_{k-3}P_{k-3}^{(1)}), \\ b'_2 = -c^{(1)}(x^2U_{k-2}P_{k-2}^{(1)}) - C_k^1 c^{(1)}(xU_{k-2}P_{k-3}^{(1)}), \\ b'_3 = -c^{(1)}(x^2U_{k-1}P_{k-2}^{(1)}) - C_k^1 c^{(1)}(xU_{k-1}P_{k-3}^{(1)}). \end{cases} \tag{3.42}$$

If $\Delta_k$ represents the determinant of the coefficients matrix of (3.41) then we have

$$\Delta_k = a'_{11}(a'_{22}a'_{33} - a'_{23}a'_{32}) - a'_{12}(a'_{21}a'_{33} - a'_{31}a'_{23}), \tag{3.43}$$

If $\Delta_k \neq 0$, then

$$\begin{cases} D_k^1 = \frac{1}{\Delta_k}\{b'_1(a'_{22}a'_{33} - a'_{23}a'_{32}) - a'_{12}(b'_2a'_{33} - b'_3a'_{23})\}, \\ F_k^1 = \frac{b'_1 - a'_{11}D_k^1}{a'_{12}}, \\ G_k^1 = \frac{b'_2 - a'_{21}D_k^1 - F_k^1 a'_{22}}{a'_{23}}. \end{cases} \tag{3.44}$$

Since, $A_k^1 = B_k^1 = 0$ and $E_k^1 = 1$, relation $B_{13new}$ becomes

$$P_k^{(1)}(x) = (C_k^1 x + D_k^1)P_{k-3}^{(1)}(x) + (x^2 + F_k^1 x + G_k^1)P_{k-2}^{(1)}(x). \tag{3.45}$$

Therefore, $B_{13new}$ can lead to a Lanczos-type algorithm.

### 3.3.2 Formula $B_{15new}$

Consider the following recurrence relationship for $k \geq 2$,

$$P_k^{(1)}(x) = (A_k^1 x^2 + B_k^1 x + C_k^1)P_{k-2} + (D_k^1 x^2 + E_k^1 x + F_k^1)P_{k-2}^{(1)}, \tag{3.46}$$

where $P_k^{(1)}(x)$, $P_{k-2}(x)$ and $P_{k-2}^{(1)}(x)$ are polynomials of degree $k$, $k-2$ and $k-2$ respectively. The constant coefficients $A_k^1$, $B_k^1$, $C_k^1$, $D_k^1$, $E_k^1$, and $F_k^1$ are to be determined by imposing the orthogonality condition (2.2) with respect to the linear function $c^{(1)}$. After multiplying equation (3.46) by $U_i$ a polynomial of exact degree $i$ and applying linear functional $c^{(1)}$ on both sides it becomes

$$c^{(1)}(U_iP_k) = A_k^1 c^{(1)}(x^2U_iP_{k-2}) + B_k^1 c^{(1)}(xU_iP_{k-2}) + C_k^1 c^{(1)}(U_iP_{k-2}) + D_k^1 c^{(1)}(x^2U_iP_{k-2}^{(1)})$$
$$+E_k^1 c^{(1)}(xU_iP_{k-2}^{(1)}) + F_k^1 c^{(1)}(U_iP_{k-2}^{(1)}). \tag{3.47}$$

Consequently, by applying (2.2), we have the relation for $i = 0, 1, ..., k-1$.

$$A_k^1 c(x^3U_iP_{k-2}) + B_k^1 c(x^2U_iP_{k-2}) + C_k^1 c(xU_iP_{k-2}) + D_k^1 c^{(1)}(x^2U_iP_{k-2}^{(1)}) + E_k^1 c^{(1)}(xU_iP_{k-2}^{(1)}) + F_k^1 c^{(1)}(U_iP_{k-2}^{(1)}) = 0 \tag{3.48}$$

The orthogonality condition is always true for $i = 0, 1, 2, ......, k-6$. For $i = k-5$, equation (3.48) gives

$$A_k^1 c(x^3U_{k-4}P_{k-2}) = 0 \quad \Rightarrow \quad c(x^3U_{k-4}P_{k-3}^{(1)}) \neq 0, \quad A_k^1 = 0.$$

Since $P_k^{(1)}$ is monic, $D_k^1 = 1$. For $i = k-4$, equation (3.48) gives

$$B_k^1 c(x^2U_{k-4}P_{k-2}) + D_k^1 c^{(1)}(x^2U_{k-4}P_{k-2}^{(1)}) = 0,$$
$$B_k^1 = -\frac{c(x^3U_{k-4}P_{k-2}^{(1)})}{c(x^2U_{k-4}P_{k-2})}. \tag{3.49}$$

For $i = k-3$, equation (3.48) gives

$$C_k^1 c(xU_{k-3}P_{k-2}) + E_k^1 c^{(1)}(xU_{k-3}P_{k-2}^{(1)}) = -c^{(1)}(x^2U_{k-3}P_{k-2}^{(1)}) - B_k^1 c(xU_{k-3}P_{k-2}). \tag{3.50}$$

For $i = k-2$, equation (3.48) gives

$$C_k^1 c(xU_{k-2}P_{k-2}) + E_k^1 c^{(1)}(xU_{k-2}P_{k-2}^{(1)}) + F_k^1 c^{(1)}(U_{k-2}P_{k-2}^{(1)}) = -c^{(1)}(x^2 U_{k-2}P_{k-2}^{(1)}) - B_k^1 c(x^2 U_{k-2}P_{k-2}). \tag{3.51}$$

For $i = k-1$, equation (3.48) gives

$$C_k^1 c(xU_{k-1}P_{k-2}) + E_k^1 c^{(1)}(xU_{k-1}P_{k-2}^{(1)}) + F_k^1 c^{(1)}(U_{k-1}P_{k-2}^{(1)}) = -c^{(1)}(x^2 U_{k-1}P_{k-2}^{(1)}) - B_k^1 c(x^2 U_{k-1}P_{k-2}). \tag{3.52}$$

Equations (3.50), (3.51) and (3.52) can be written as

$$\begin{cases} a'_{11}C_k^1 + a'_{12}E_k^1 = b'_1, \\ a'_{21}C_k^1 + a'_{22}E_k^1 + a'_{23}F_k^1 = b'_2, \\ a'_{31}C_k^1 + a'_{32}E_k^1 + a'_{33}F_k^1 = b'_3. \end{cases} \tag{3.53}$$

Where $a'_{11}, a'_{12}, a'_{21}, a'_{22}, a'_{23}, a'_{31}, a'_{32}, a'_{33}$, are the coefficients of $C_k^1$, $E_k^1$, and $F_k^1$. Suppose $b'_1$, $b'_2$, and $b'_3$ are the corresponding right hand side terms of these equations.

$$\begin{cases} b'_1 = -c(x^3 U_{k-3}P_{k-2}^{(1)}) - B_k^1 c(xU_{k-3}P_{k-2}), \\ b'_2 = -c(x^3 U_{k-2}P_{k-2}^{(1)}) - B_k^1 c(x^2 U_{k-2}P_{k-2}), \\ b'_3 = -c(x^3 U_{k-1}P_{k-2}^{(1)}) - B_k^1 c(x^2 U_{k-1}P_{k-2}). \end{cases} \tag{3.54}$$

If $\Delta_k^1$ represents the determinant of the coefficients matrix of (3.53) then we have

$$\Delta_k^1 = a'_{11}(a'_{22}a'_{33} - a'_{23}a'_{32}) - a'_{12}(a'_{21}a'_{33} - a'_{31}a'_{23}).$$

If $\Delta_k \neq 0$, then

$$\begin{cases} C_k^1 = \frac{b'_1(a'_{22}a'_{33} - a'_{23}a'_{32}) - a'_{12}(b'_2 a'_{33} - b'_3 a'_{23})}{\Delta_k^1}, \\ E_k^1 = \frac{b'_1 - C_k^1 a'_{11}}{a'_{12}}, \\ F_k^1 = \frac{b'_2 - a'_{21}C_k^1 - E_k^1 a'_{22}}{a'_{23}}. \end{cases} \tag{3.55}$$

Since, $A_k^1 = 0$ and $D_k^1 = 1$, relation $B_{15new}$ becomes

$$P_k^{(1)}(x) = (B_k^1 x + C_k^1)P_{k-2}(x) + (x^2 + E_k^1 x + F_k^1)P_{k-2}^{(1)}(x). \tag{3.56}$$

This means $B_{15new}$ can lead to the implementation of a Lanczos-type algorithm.

### 3.3.3 Formula $B_{16new}$

Consider the following recurrence relationship for $k \geq 2$,

$$P_k^{(1)}(x) = (A_k^1 x^2 + B_k^1 x + C_k^1)P_{k-2}^{(1)} + (D_k^1 x + E_k^1)P_{k-1}, \tag{3.57}$$

where $P_k^{(1)}(x)$, $P_{k-2}^{(1)}(x)$ and $P_{k-1}(x)$ are polynomials of degree $k$, $k-2$ and $k-1$ respectively. The constant coefficients $A_k^1$, $B_k^1$, $C_k^1$, $D_k^1$, and $E_k^1$, are to be determined by imposing the orthogonality condition (2.2) with respect to the linear function $c^{(1)}$. After multiplying (3.57) by $U_i$ a polynomial of exact degree $i$ and applying linear functional $c^{(1)}$ on both sides it becomes

$$c^{(1)}(U_i P_k^{(1)}) = A_k^1 c^{(1)}(x^2 U_i P_{k-2}^{(1)}) + B_k^1 c^{(1)}(x U_i P_{k-2}^{(1)}) + C_k^1 c^{(1)}(U_i P_{k-2}^{(1)}) + D_k^1 c^{(1)}(x U_i P_{k-1}) + E_k^1 c^{(1)}(U_i P_{k-1}). \tag{3.58}$$

Consequently, by applying (2.2), we have the relation for $i = 0, 1, ..., k-1$

$$A_k^1 c^{(1)}(x^2 U_i P_{k-2}^{(1)}) + B_k^1 c^{(1)}(x U_i P_{k-2}^{(1)}) + C_k^1 c^{(1)}(U_i P_{k-2}^{(1)}) + D_k^1 c^{(1)}(x U_i P_{k-1}) + E_k^1 c^{(1)}(U_i P_{k-1}) = 0,$$

$$A_k^1 c^{(1)}(x^2 U_i P_{k-2}^{(1)}) + B_k^1 c^{(1)}(x U_i P_{k-2}^{(1)}) + C_k^1 c^{(1)}(U_i P_{k-2}^{(1)}) + D_k^1 c(x^2 U_i P_{k-1}) + E_k^1 c(x U_i P_{k-1}) = 0. \tag{3.59}$$

Equation (3.59) is always true $i = 0, 1, 2, ..., k-5$.

For $i = k-4$, equation (3.59) gives

$$A_k^1 c^{(1)}(x^2 U_{k-4} P_{k-2}^{(1)}) = 0 \quad \Rightarrow \quad c^{(1)}(x^2 U_{k-4} P_{k-2}^{(1)}) \neq 0, \quad A_k^1 = 0.$$

Since $P_k^{(1)}(x)$ is monic, therefore $D_k^1 a_{k-1} = 1, \Rightarrow D_k^1 = \frac{1}{a_{k-1}}$

For $i = k-3$, equation (3.59) gives

$$B_k^1 c^{(1)}(x U_{k-3} P_{k-2}^{(1)}) + D_k^1 c^{(1)}(x U_{k-3} P_{k-1}) = 0, \Rightarrow B_k^1 = -\frac{D_k^1 c^{(1)}(x U_{k-3} P_{k-1})}{c^{(1)}(x U_{k-3} P_{k-2}^{(1)})}.$$

For $i = k-2$, equation (3.59) gives

$$C_k^1 c(x U_{k-2} P_{k-2}^{(1)}) + E_k^1 c(x U_{k-2} P_{k-1}) = -B_k^1 c(x^2 U_{k-2} P_{k-2}^{(1)}) - D_k^1 c(x^2 U_{k-2} P_{k-1}). \tag{3.60}$$

For $i = k-1$, equation (3.59) gives

$$C_k^1 c(xU_{k-1}P_{k-2}^{(1)}) + E_k^1 c(xU_{k-1}P_{k-1}) = -B_k^1 c(x^2U_{k-1}P_{k-2}^{(1)}) - D_k^1 c(x^2U_{k-1}P_{k-1}). \quad (3.61)$$

Equations (3.60) and (3.61) can be written as

$$\begin{cases} a'_{11}C_k^1 + a'_{12}E_k^1 = b'_1, \\ a'_{21}C_k^1 + a'_{22}E_k^1 = b'_2 \end{cases} \quad (3.62)$$

Where $a'_{11}$, $a'_{12}$, $a'_{21}$, and $a'_{22}$, are the coefficients of $C_k^1$, and $E_k^1$, and suppose $b'_1$, and $b'_2$ are the corresponding right hand side terms of these equations.

$$\begin{cases} b'_1 = -B_k^1 c(x^2U_{k-2}P_{k-2}^{(1)}) - D_k^1 c(x^2U_{k-2}P_{k-1}), \\ b'_2 = -B_k^1 c(x^2U_{k-1}P_{k-2}^{(1)}) - D_k^1 c(x^2U_{k-1}P_{k-1}). \end{cases} \quad (3.63)$$

If $\Delta_k$ represents the determinant of the coefficients matrix of (3.62) then we have

$$\Delta_k = a'_{11}a'_{22} - a'_{12}a'_{21}.$$

If $\Delta_k \neq 0$, then

$$\begin{cases} D_k^1 = \frac{1}{a_{k-1}}, \\ B_k^1 = -\frac{D_k^1 c(x^2P_{k-3}P_{k-1})}{c(x^2P_{k-3}P_{k-2}^{(1)})}, \\ C_k^1 = \frac{b'_1a'_{22} - b'_2a'_{12}}{\Delta_k}, \\ E_k^1 = \frac{b'_2a'_{11} - b'_1a'_{12}}{\Delta_k}. \end{cases} \quad (3.64)$$

Since, $A_k^1 = 0$, relation $B_{16new}$ becomes

$$P_k^{(1)}(x) = (B_k^1x + C_k^1)P_{k-2}^{(1)}(x) + (D_k^1x + E_k^1)P_{k-1}(x). \quad (3.65)$$

This means $B_{16new}$ can lead to the implementation of a Lanczos-type algorithm.

## 3.4 Lanczos-type Algorithms for the Choice of $U_i(x) = P_i(x)$

In this chapter, we have derived new FOPs based recurrence formulae. Now we derive Lanczos-type algorithm which are based on these formulae. If we write $\mathbf{r}_k = P_k(x)\mathbf{r}_0$, $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$ and $\mathbf{z}_k = P_k^{(1)}(x)\mathbf{r}_0$, the formulae $A_i$ provide expressions for $\mathbf{r}_k$ and $\mathbf{x}_k$, and the formulae $B_j$ help to find $\mathbf{z}_k$, recursively.

### 3.4.1 $A_{16new}/B_{15new}$ Based Lanczos-type Algorithm

From relation $A_{16new}$ of subsection 3.2.2, the equation (3.25), after replacing $x$ by $A$. Since $\mathbf{r}_k = P_k(A)\mathbf{r}_0$, we have

$$\begin{cases} \mathbf{r}_k = \mathbf{r}_{k-2} + B_k A\mathbf{r}_{k-2} + D_k A^2\mathbf{z}_{k-2} + E_k A\mathbf{z}_{k-2}, \\ \tilde{\mathbf{r}}_k = \tilde{\mathbf{r}}_{k-2} + B_k A^T\tilde{\mathbf{r}}_{k-2} + D_k(A^T)^2\tilde{\mathbf{z}}_{k-2} + E_k A^T\tilde{\mathbf{z}}_{k-2}. \end{cases} \tag{3.66}$$

Using $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, we get

$$\mathbf{x}_k = \mathbf{x}_{k-2} - B_k\mathbf{r}_{k-2} - D_k A\mathbf{z}_{k-2} - E_k\mathbf{z}_{k-2}. \tag{3.67}$$

The equations (3.66) and (3.67) with all coefficients involved have been derived as (3.24) in subsection 3.2.2, are valid for $k \geq 2$. We have to calculate $\mathbf{r}_1$ and $\mathbf{x}_1$ differently as in equations (2.138).

If we set,

$$\begin{cases} \mathbf{r}_k = P_k\mathbf{r}_0, \;\; \tilde{\mathbf{r}}_k = P_k(A^T)\mathbf{y}, \\ \mathbf{z}_k = P_k^{(1)}(A)\mathbf{r}_0, \;\; \tilde{\mathbf{z}}_k = P_k^{(1)}(A^T)\tilde{\mathbf{z}}_0. \end{cases} \tag{3.68}$$

Now, for $U_i(x) = P_i(x)$. Therefore, the rest of the coefficients can be written explicitly as follow;

$a_{11} = (\tilde{\mathbf{r}}_{k-3}, A\mathbf{r}_{k-2})$, $a_{12} = (\tilde{\mathbf{r}}_{k-3}, A^2\mathbf{z}_{k-2})$, $a_{21} = (\tilde{\mathbf{r}}_{k-2}, A\mathbf{r}_{k-2})$, $a_{22} = (\tilde{\mathbf{r}}_{k-2}, A^2\mathbf{z}_{k-2})$,

$a_{23} = (\tilde{\mathbf{r}}_{k-2}, A\mathbf{z}_{k-2})$, $a_{31} = (\tilde{\mathbf{r}}_{k-2}, A\mathbf{r}_{k-1})$, $a_{32} = (\tilde{\mathbf{r}}_{k-1}, A^2\mathbf{z}_{k-2})$, $a_{33} = (\tilde{\mathbf{r}}_{k-1}, A\mathbf{z}_{k-2})$.

$b_1 = 0$, $b_2 = -(\tilde{\mathbf{r}}_{k-2}, \mathbf{r}_{k-2})$, $b_3 = -c(P_{k-1}P_{k-2}) = 0$.

From formula $B_{15new}$ of subsection 3.3.2, equation (3.56), after replacing $x$ by $A$. Since $\mathbf{z}_k = P_k^{(1)}(A)\mathbf{r}_0$, we have

$$\begin{cases} \mathbf{z}_k = B_k A\mathbf{r}_{k-2} + C_k\mathbf{r}_{k-2} + A^2\mathbf{z}_{k-2} + E_k A\mathbf{z}_{k-2} + F_k\mathbf{z}_{k-2}, \\ \tilde{\mathbf{z}}_k = B_k A^T\tilde{\mathbf{r}}_{k-2} + C_k\tilde{\mathbf{r}}_{k-2} + (A^T)^2\tilde{\mathbf{z}}_{k-2} + E_k A^T\tilde{\mathbf{z}}_{k-2} + F_k\tilde{\mathbf{z}}_{k-2}. \end{cases} \tag{3.69}$$

The equations (3.69) with all coefficients involved have been derived as (3.49) and (3.55) in subsection 3.3.2, are valid for $k \geq 2$.

Now, for $U_i(x) = P_i(x)$, if we set, $\mathbf{r}_k = P_k(A)\mathbf{r}_0$, $\tilde{\mathbf{r}}_k = P_k(A^T)\mathbf{y}$, and $\mathbf{z}_k = P_k^{(1)}(A)\mathbf{r}_0$

$a'_{11} = (\tilde{\mathbf{r}}_{k-3}, A\mathbf{r}_{k-2})$, $a'_{12} = (\tilde{\mathbf{r}}_{k-3}, A^2\mathbf{z}_{k-2})$, $a'_{21} = (\tilde{\mathbf{r}}_{k-2}, A\mathbf{r}_{k-2})$, $a'_{22} = (\tilde{\mathbf{r}}_{k-2}, A^2\mathbf{z}_{k-2})$,

$a'_{23} = (\tilde{\mathbf{r}}_{k-2}, A\mathbf{z}_{k-2})$, $a'_{31} = (\tilde{\mathbf{r}}_{k-2}, A\mathbf{r}_{k-1})$, $a_{32} = (\tilde{\mathbf{r}}_{k-1}, A^2\mathbf{z}_{k-2})$, $a'_{33} = (\tilde{\mathbf{r}}_{k-1}, A\mathbf{z}_{k-2})$.

$b_1 = -(\tilde{\mathbf{r}}_{k-3}, A^3\mathbf{z}_{k-2}) - B_{k+1}(\tilde{\mathbf{r}}_{k-3}, A^2\mathbf{r}_{k-2})$, $b_2 = -(\tilde{\mathbf{r}}_{k-2}, A^3\mathbf{z}_{k-2}) - B_{k+1}(\tilde{\mathbf{r}}_{k-2}, A^2\mathbf{r}_{k-2})$,

$b_3 = -(\tilde{\mathbf{r}}_{k-1}, A^3\mathbf{z}_{k-2}) - B_{k+1}(\tilde{\mathbf{r}}_{k-1}, A^2\mathbf{r}_{k-2})$.

After gathering together all these formulae, we finally have the Lanczos algorithm based on $A_{16new}$ and $B_{15new}$.

### 3.4.2 $A_{16new}/B_{16new}$ Based Lanczos-type Algorithm

From equations (3.66), and (3.67), we have

$$\begin{cases} \mathbf{r}_k = \mathbf{r}_{k-2} + B_k A\mathbf{r}_{k-2} + D_k A^2\mathbf{z}_{k-2} + E_k A\mathbf{z}_{k-2}, \\ \tilde{\mathbf{r}}_k = \tilde{\mathbf{r}}_{k-2} + B_k A^T\tilde{\mathbf{r}}_{k-2} + D_k (A^T)^2\tilde{\mathbf{z}}_{k-2} + E_k A^T\tilde{\mathbf{z}}_{k-2}, \\ \mathbf{x}_k = \mathbf{x}_{k-2} - B_k\mathbf{r}_{k-2} - D_k A\mathbf{z}_{k-2} - E_k\mathbf{z}_{k-2}. \end{cases} \tag{3.70}$$

The equations (3.70) with all coefficients involved have been derived as (3.24) in subsection 3.2.2, are valid for $k \geq 2$. We have to calculate $\mathbf{r}_1$ and $\mathbf{x}_1$ differently as in equations (2.138). From formula $B_{16new}$ of subsection 3.3.3, equation (3.65), after replacing $x$ by $A$. Since $\mathbf{z}_k = P_k^{(1)}(A)\mathbf{r}_0$, we have

$$\begin{cases} \mathbf{z}_k = B_k A\mathbf{z}_{k-2} + C_k\mathbf{z}_{k-2} + D_k A\mathbf{r}_{k-1} + E_k\mathbf{r}_{k-1}. \\ \tilde{\mathbf{z}}_k = B_k A^T\tilde{\mathbf{z}}_{k-2} + C_k\tilde{\mathbf{z}}_{k-2} + D_k A^T\tilde{\mathbf{r}}_{k-1} + E_k\tilde{\mathbf{r}}_{k-1}. \end{cases} \tag{3.71}$$

The equations (3.71) with all coefficients involved have been derived as (3.64) in subsection 3.3.3, are valid for $k \geq 2$. Therefore, we need to find $\mathbf{z}_1$ as in Eq (2.146). Since $D_k^1 = \frac{1}{a_{k-1}}$ is defined by $P_{k-1} = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + ... + 1$,, and $a_k = Da_{k-1}$, $a_{k-1} = D_{k-1}k_{k-2}$, therefore, $D_k^1 = \frac{D_{k-1}^1}{D_{k-1}}$.

Now, for $U_i(x) = P_i(x)$, using Eq (3.68) the rest of the coefficients can be written explicitly as follows;

$a'_{11} = (\tilde{\mathbf{r}}_{k-2}, A\mathbf{z}_{k-2}), \quad a'_{12} = (\tilde{\mathbf{r}}_{k-2}, A\mathbf{r}_{k-1}), \quad a'_{21} = (\tilde{\mathbf{r}}_{k-1}, A\mathbf{z}_{k-2}), \quad a'_{22} = (\tilde{\mathbf{r}}_{k-1}, A\mathbf{r}_{k-1})$

$b'_1 = -B_k(\tilde{\mathbf{r}}_{k-2}, A^2\mathbf{z}_{k-2}) - D_k(\tilde{\mathbf{r}}_{k-2}, A^2\mathbf{r}_{k-1}), \; b'_2 = -B_k(\tilde{\mathbf{r}}_{k-1}, A^2\mathbf{z}_{k-2}) - D_k(\tilde{\mathbf{r}}_{k-1}, A^2\mathbf{r}_{k-1})$

After gathering together all these formulae, we finally have the Lanczos algorithm based on $A_{16new}$ and $B_{16new}$.

### 3.4.3 $A_{19new}/B_{15new}$ Based Lanczos-type Algorithm

From formula $A_{19new}$ of subsection 3.2.3, the equation (3.34), after replacing $x$ by $A$. Since $\mathbf{r}_k = P_k(A)\mathbf{r}_0$, we have

$$\begin{cases} \mathbf{r}_k = \mathbf{r}_{k-1} + D_k A\mathbf{r}_{k-1} + B_k A\mathbf{z}_{k-2}, \\ \tilde{\mathbf{r}}_k = \tilde{\mathbf{r}}_{k-1} + D_k A^T\tilde{\mathbf{r}}_{k-1} + B_k A^T\tilde{\mathbf{z}}_{k-2}. \end{cases} \tag{3.72}$$

Using $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, we get

$$\mathbf{x}_k = \mathbf{x}_{k-1} - B_k\mathbf{z}_{k-2} - D_k\mathbf{r}_{k-1}. \tag{3.73}$$

The equations (3.72) and (3.73) with all coefficients involved having been derived as (3.33) in subsection 3.2.3, are valid for $k \geq 2$. However, we have to calculate $\mathbf{r}_1$, $\mathbf{x}_1$ differently as in (2.138) and $\tilde{\mathbf{r}}_1$ from equations (2.138) we have

$$\tilde{\mathbf{r}}_1 = \tilde{\mathbf{r}}_0 - \frac{c_0}{c_1}A^T\tilde{\mathbf{r}}_0. \tag{3.74}$$

Now, for $U_i(x) = P_i(x)$, using Eq (3.68) the rest of the coefficients can be written explicitly as follow;

$a_{11} = (\tilde{\mathbf{r}}_{k-2}, A\mathbf{z}_{k-2}), \quad a_{12} = (\tilde{\mathbf{r}}_{k-2}, A\mathbf{r}_{k-1}), \quad a_{21} = (\tilde{\mathbf{r}}_{k-1}, A\mathbf{z}_{k-2}), \quad a_{22} = (\tilde{\mathbf{r}}_{k-1}, A\mathbf{r}_{k-1})$

$b_1 = 0, \quad b_2 = -(\tilde{\mathbf{r}}_{k-1}, \mathbf{r}_{k-1})$.

From equation (3.68) in subsection (3.4.1), we have for $B_{15new}$,

$$\begin{cases} \mathbf{z}_k = B_k^1 A\mathbf{r}_{k-2} + C_k^1 \mathbf{r}_{k-2} + A^2 \mathbf{z}_{k-2} + E_k^1 A\mathbf{z}_{k-2} + F_k^1 \mathbf{z}_{k-2}. \\ \tilde{\mathbf{z}}_k = B_k^1 A^T \tilde{\mathbf{r}}_{k-2} + C_k^1 \tilde{\mathbf{r}}_{k-2} + (A^T)^2 \tilde{\mathbf{z}}_{k-2} + E_k^1 A^T \tilde{\mathbf{z}}_{k-2} + F_k^1 \tilde{\mathbf{z}}_{k-2}. \end{cases} \quad (3.75)$$

The equations (3.75) with all coefficients involved already derived as (3.49) and (3.55) in subsection 3.3.2, are valid for $k \geq 2$. Therefore, we need to find $\mathbf{z}_1$, and $\tilde{\mathbf{z}}_1$ by alternative ways as in (2.146), (2.147) and (2.148) of subsection 2.4.2.

$$\begin{cases} \tilde{\mathbf{z}}_1 = A\tilde{\mathbf{z}}_0 - \frac{c_2}{c_1}\tilde{\mathbf{z}}_0, \\ \tilde{\mathbf{z}}_2 = A^2\tilde{\mathbf{z}}_0 - \mu A\tilde{\mathbf{z}}_0 + \nu\tilde{\mathbf{z}}_0, \\ \tilde{\mathbf{z}}_3 = A^3\tilde{\mathbf{z}}_0 - \eta' A^2\tilde{\mathbf{z}}_0 + \mu' A\tilde{\mathbf{z}}_0 - \nu'\tilde{\mathbf{z}}_0. \end{cases} \quad (3.76)$$

We finally have Algorithm 9, after gathering together all these formulae.

**Algorithm 9** Lanczos-type Algorithm based on relations $A_{19new}/B_{15new}$

**Input:** $A$ an $n \times n$ matrix, $b$ an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $||\mathbf{r}_k||$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, such that $y \neq 0$ and the tolerance $\varepsilon$ to $1.0E-13$.
Set $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\mathbf{y}_0 = \mathbf{y}$, $\mathbf{z}_0 = \mathbf{r}_0$, $\tilde{\mathbf{z}}_0 = \mathbf{y}$, $\tilde{\mathbf{r}}_0 = \mathbf{y}$.
**Compute:**
$c_0$ and $c_1$, as in (1.23b),
$\mathbf{r}_1$, $\mathbf{x}_1$, as in (2.138), $\tilde{\mathbf{r}}_1$, as in (3.74)
$\mathbf{z}_1$, as in (2.146), $\tilde{\mathbf{z}}_1$, $\tilde{\mathbf{z}}_2$, $\tilde{\mathbf{z}}_3$ as in (3.76)
$k = 2$;
**While** $||\mathbf{r}_k|| > \varepsilon$ **do**
$B_k$, $D_k$ as in subsection (3.33),
$\mathbf{r}_k = \mathbf{r}_{k-1} + D_k A\mathbf{r}_{k-1} + B_k A\mathbf{z}_{k-1}$,
$\tilde{\mathbf{r}}_k = \tilde{\mathbf{r}}_{k-1} + D_k A'\tilde{\mathbf{r}}_{k-1} + B_k A'\tilde{\mathbf{z}}_{k-1}$,
$\mathbf{x}_k = \mathbf{x}_{k-1} - D_k\mathbf{r}_{k-1} - B_k A\mathbf{z}_{k-2}$.
$B_k'$, as in (3.49),
$C_k'$, $E_k'$, $F_k'$, as in (3.55),
$\mathbf{z}_k = B_k' A\mathbf{r}_{k-2} + C_k'\mathbf{r}_{k-2} + A^2\mathbf{z}_{k-2} + E_k' A\mathbf{z}_{k-2} + F_k'\mathbf{z}_{k-2}$,
$\tilde{\mathbf{z}}_k = B_k' A'\tilde{\mathbf{r}}_{k-2} + C_k'\tilde{\mathbf{r}}_{k-2} + A'^2\tilde{\mathbf{z}}_{k-2} + E_k' A'\tilde{\mathbf{z}}_{k-2} + F_k'\tilde{\mathbf{z}}_{k-2}$.
$k = k + 1$.
**EndWhile**
Obtain the approximate solution as well as the residual norm.
$\text{sol}_{last} = \mathbf{x}_k$,
$\text{norm}_{last} = ||\mathbf{r}_k||$.
**Stop**.

### 3.4.4 $A_{19new}/B_{16new}$ Based Lanczos-type Algorithm

From equation (3.72), and (3.73) in subsection 3.4.3, we have for $A_{19new}$

$$\begin{cases} \mathbf{r}_k = \mathbf{r}_{k-1} + D_k A \mathbf{r}_{k-1} + B_k A \mathbf{z}_{k-2}, \\ \tilde{\mathbf{r}}_k = \tilde{\mathbf{r}}_{k-1} + D_k A^T \tilde{\mathbf{r}}_{k-1} + B_k A^T \tilde{\mathbf{z}}_{k-2}, \\ \mathbf{x}_k = \mathbf{x}_{k-1} - B_k \mathbf{z}_{k-2} - D_k \mathbf{r}_{k-1}. \end{cases} \tag{3.77}$$

The Eqs (3.77) with all coefficients involved already derived as Eq (3.33) in subsection 3.2.3, are valid for $k \geq 2$. We have to calculate $\mathbf{r}_1$ and $\mathbf{x}_1$ differently as Eq (2.138). From Eqs (3.71) in subsection 3.4.2, we have

$$\begin{cases} \mathbf{z}_k = B_k^1 A \mathbf{z}_{k-2} + C_k^1 \mathbf{z}_{k-2} + D_k^1 A \mathbf{r}_{k-1} + E_k^1 \mathbf{r}_{k-1}, \\ \tilde{\mathbf{z}}_k = B_k^1 A^T \tilde{\mathbf{z}}_{k-2} + C_k^1 \tilde{\mathbf{z}}_{k-2} + D_k^1 A^T \tilde{\mathbf{r}}_{k-1} + E_k^1 \tilde{\mathbf{r}}_{k-1}. \end{cases} \tag{3.78}$$

Similarly, Eqs (3.78) with all coefficients involved already derived in subsection 3.3.3, are valid for $k \geq 2$. Therefore, we only need to find $\mathbf{z}_1$, as Eq (2.146) and $\tilde{\mathbf{z}}_1$ as Eq (3.76)

### 3.4.5 Numerical Results of $A_{19new}/B_{15new}$

The algorithms are coded in Matlab R2014b and run on a PC under Microsoft Windows 7 Enterprise, with 16.00GB RAM, and processor Intel(R) Core(TM) i5-3570 CPU 3.40GHz. Experimental results are recorded in the Table 3.1 for different size problems ranging from 10 to 5000 of Baheux-type problems [3, 33]. Experimental results on instances of problem $Ax = b$ with $A$ refer in section 2.5 are recorded in the following Table 3.1. The stoping criterion is the norm of residual $\|r_k\| = tol = 1.0000E-13$.

**Algorithm 10** Restarting Lanczos-type Algorithm based on relations $A_{19new}/B_{15new}$

Run Algorithm 9 for a fixed number of iterations $k$ or until it halts;
Obtain the solution $sol_{last} = \mathbf{x}_k$ as well as the residual norm $norm_{last} = ||\mathbf{r}_k||$.
**While** $||\mathbf{r}_k|| > \varepsilon$ **do**
  initialize it with the current iterate of the algorithm run,
  $\mathbf{x} = sol_{last}$,
  $\mathbf{y} = \mathbf{b} - A\mathbf{x}$.
  Run Algorithm 9 for a fixed number of iterations $k$
**EndWhile**
Obtain the optimal solution as well as the optimal residual norm as follows
$sol_{optimal} = \mathbf{x}_k$
$norm_{optimal} = ||\mathbf{r}_k||$.
**Stop**.

**Table 3.1:** *Results of Algorithm 9 and Algorithm 10 on Baheux-type problems for $\delta = 0$*

| Dim of Prob | Algorithm 9 | | Algorithm 10 | |
|---|---|---|---|---|
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | sec | $\|\|r_k\|\|$ | sec |
| 10 | 1.5145E-16 | 9.8426E-01 | 1.5145E-16 | 9.6844E-01 |
| 50 | 1.5310E-14 | 8.8283E-01 | 1.5310E-14 | 9.1245E-01 |
| 100 | 7.0504E-15 | 9.8697E-01 | 7.0504E-15 | 9.3030E-01 |
| 200 | NaN | | 8.1359E-14 | 1.3221E+00 |
| 500 | NaN | | 9.3504E-14 | 1.0537E+01 |
| 1000 | NaN | | 9.1007E-14 | 7.9361E+01 |
| 5000 | NaN | | 8.6604E-14 | 7.5275E+03 |
| 10000 | NaN | | 8.5147E-14 | 5.2613E+04 |

Table 3.1 lists the results obtained from computations with Algorithm 9 $(A_{19}/B_{15})_{new}$, and its restart version Algorithm 10. It is clear from the results that the Lanczos-type algorithm suffers from breakdown. It is due to a division by zero that can not be avoided when computing the coefficients of those recurrence relations based on $P_k(x)$ and $P_k^{(1)}(x)$. The coefficients of different recurrence relations between orthogonal polynomials consist of ratios of scalar products. Some of the scalar products in the denominator are as small as E-14, which causes the breakdown and the algorithms have to be stopped. Secondly,

causes of breakdown may be due to the non-existence of some of the FOPs involved in the recurrence relations. Restarting is used to avoid the problem. This strategy either stops the Lancozs-type algorithm pre-emptively and restarts it with some iterate or waits until breakdown occurs and then restarts from the last iterate found.

## 3.5 Summary

The focus of this chapter was on obtaining the recurrence relations between FOPs taking into consideration the common family of auxiliary polynomials $U_i(x)$. This relation for $U_i(x) = x^i$ [33] is then explained concisely. Following this, the expressions for the coefficients of this polynomial are derived for a new choice of $U_i(x) = P_i(x)$. The relations $A_i/B_j$ [33] are also recalled for the same choice of the auxiliary polynomials $U_i(x) = P_i(x)$ or $U_i(x) = P_i^{(1)}(x)$. It should be noted that these Lanczos-type of algorithms suffers from breakdown. This issue is going to be addressed in the next chapter.

# Chapter 4

# Monitoring breakdown issue in Lanczos-type algorithms

## 4.1 Introduction

Because every algorithms relies on different recurrence relations between different FOPs, it is difficult to generate a test for monitoring the components that cause breakdown which is valid for all Lanczos-type algorithms. Every algorithm, therefore will have its own test. This is the best one can do at the moment. It is worth noting that for a given Lanczos-type algorithm the test works well and prevent the algorithm from breaking down.

## 4.2 Recalling some existing Lanczos-type algorithms

We revisit some established Lanczos-type algorithms such as $A_{12}$ [33], Orthores, Orthodir and Orthomin as mentioned in [4].

### 4.2.1 Lanczos-type algorithm based on relation $A_{12}$

Consider the recurrence relationship for $k \geq 3$,

$$P_k(x) = A_k\Big\{(x^2 + B_k x + C_k)P_{k-2} + (D_k x^3 + E_k x^2 + F_k x + G_k)P_{k-3}\Big\}, \tag{4.1}$$

where $P_k(x)$, $P_{k-2}(x)$ and $P_{k-3}(x)$ are polynomials of degree $k$, $k-2$ and $k-3$ respectively. The constant coefficients $A_k$, $B_k$, $C_k$, $D_k$, $E_k$,$F_k$, and $G_k$ are determined by the normalization condition $P_k(0) = 1$ and imposing the orthogonality condition (2.1). For the detailed derivation, the identification of the coefficients and the algorithm itself, please refer to [33]. From the above we immediately obtain

$$P_k(x) = A_k\Big\{(x^2 + B_k x + C_k)P_{k-2} + (F_k x + G_k)P_{k-3}\Big\}. \tag{4.2}$$

Their coefficients are estimated as $D_k = 0$ and $E_k = 0$. If $\Delta_k \neq 0$, then

$$B_k = \frac{b_1(a_{22}a_{33} - a_{32}a_{23}) + a_{13}(b_2 a_{32} - b_3 a_{22})}{\Delta_k}, \tag{4.3}$$

where $\Delta_k = a_{11}(a_{22}a_{33} - a_{32}a_{23}) + a_{13}(a_{21}a_{32} - a_{31}a_{22})$,

$$F_k = -\frac{c(x^{k-2}P_{k-2})}{c(x^{k-3}P_{k-3})}.$$

$$\begin{cases} G_k = \frac{b_1 - a_{11}B_k}{a_{13}}, \\ C_k = \frac{b_2 - a_{21}B_k - a_{23}G_k}{a_{22}}, \\ A_k = \frac{1}{C_k + G_k}. \end{cases} \tag{4.4}$$

Since $\mathbf{r}_k = P_k(A)\mathbf{r}_0$, the equation (4.2), after replacing $x$ by $A$ and using $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, we get

$$\begin{cases} \mathbf{r}_k = A_k\Big\{(A^2 + B_k A + C_k)\mathbf{r}_{k-2} + (F_k A + G_k)\mathbf{r}_{k-3}\Big\}, \\ \mathbf{x}_k = A_k\Big\{C_k\mathbf{x}_{k-2} + G_k\mathbf{x}_{k-3} - (A\mathbf{r}_{k-2} + B_k\mathbf{r}_{k-2} + F_k)\mathbf{r}_{k-3}\Big\}. \end{cases} \tag{4.5}$$

Equations (4.5) define a Lanczos-type algorithm. Now, we have to find the expressions of the coefficients $A_k$, $B_k$, $C_k$, $F_k$, and $G_k$ appearing in them. We know that Therefore, we can write using Eq (2.136) we get

$$F_k = -\frac{(\mathbf{y}_{k-2}, \mathbf{r}_{k-2})}{(\mathbf{y}_{k-3}, \mathbf{r}_{k-3})}. \tag{4.6}$$

The rest of the coefficients can be written explicitly as follows:

$a_{11} = (\mathbf{y}_{k-2}, \mathbf{r}_{k-2})$, $a_{12} = 0$, $a_{13} = (\mathbf{y}_{k-3}, \mathbf{r}_{k-3})$,

$a_{21} = (\mathbf{y}_{k-1}, \mathbf{r}_{k-2})$, $a_{22} = a_{11}$, $a_{23} = (\mathbf{y}_{k-2}, \mathbf{r}_{k-3})$,

$a_{31} = (\mathbf{y}_{k}, \mathbf{r}_{k-2})$, $a_{32} = a_{21}$, $a_{33} = (\mathbf{y}_{k-1}, \mathbf{r}_{k-3})$,

$b_1 = -a_{21} - F_k a_{23}$, $\quad b_2 = -a_{31} - F_k a_{33}$, $\quad b_3 = -s - F_k t$,

where $s = (\mathbf{y}_{k+1}, \mathbf{r}_{k-2})$, $\; t = (\mathbf{y}_{k}, \mathbf{r}_{k-3})$

We finally have the following algorithm after gathering all these formulae [33].

**Algorithm 11** Lanczos-type Algorithm based on relation $A_{12}$

**Input:** $A$ an $n \times n$ matrix, $\mathbf{b}$ an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $\|\mathbf{r}_k\|$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, such that $\mathbf{y} \neq 0$ and the tolerance $\varepsilon$ to $1E-13$.
Set $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$; $\quad \mathbf{y}_0 = \mathbf{y}$;
**Compute:**
$c_0, c_1, c_2, c_3$; as in (1.23b)
$\mathbf{r}_1$, and $\mathbf{x}_1$, as in (2.138), [33]
$\mathbf{r}_2$ and $\mathbf{x}_2$ and (2.139), [33]
$k = 2$;
**While** $\|\mathbf{r}_k\| > \varepsilon$ **do**
$\mathbf{y}_{k+1} = A^T \mathbf{y}_k$;
$B_k$ as in (4.3),
$A_k$, $C_k$, and $G_k$, as in (4.4),
$F_k$ as in (4.6).
$\mathbf{r}_k$ and $\mathbf{x}_k$ as in (4.5).
$k = k + 1$;
**EndWhile**
Obtain the approximate solution as well as the residual norm.
$\text{sol}_{last} = \mathbf{x}_k$;
$\text{norm}_{last} = \|\mathbf{r}_k\|$;
**Stop**.

### 4.2.2 Lanczos-type Algorithm Based on Relation $A_4$

Algorithm $A_4$ is well-known as the **Orthores** algorithm [4]. Let us now consider the recurrence relation on which it is based. It written can be as

$$P_k(x) = A_k\Big\{(x + B_k)P_{k-1} + (C_k x^2 + D_k x + E_k)P_{k-2}\Big\}, \tag{4.7}$$

where $P_k(x)$, $P_{k-1}(x)$ and $P_{k-2}(x)$ are polynomials of degree $k$, $k-1$ and $k-2$ respectively. The constant coefficients $A_k$, $B_k$, $C_k$, $D_k$, and $E_k$, are determined by the normalization condition $P_k(0) = 1$ and imposing the orthogonality condition (2.1). For the detailed derivation, the identification of the coefficients and the algorithm itself, please refer to [4].

From the above we immediately obtain

$$P_k(x) = A_k\Big\{(x + B_k x)P_{k-1} + E_k P_{k-2}\Big\}. \tag{4.8}$$

Their coefficients are estimated as $C_k = 0$ and $D_k = 0$,

$$\begin{cases} E_k = -\frac{c(x^{k-1}P_{k-1})}{c(x^{k-2}P_{k-2})}, \\ B_k = \frac{-c(x^k P_{k-1}) - E_k c(x^{k-1}P_{k-2})}{c(x^{k-1}P_{k-1})}, \\ A_k = \frac{1}{B_k + E_k}. \end{cases} \tag{4.9}$$

Since $\mathbf{r}_k = P_k(A)\mathbf{r}_0$, the equation (4.8), after replacing $x$ by $A$ and using $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, we get

$$\begin{cases} \mathbf{r}_k = A_k\Big\{(A\mathbf{r}_{k-1} + B_k\mathbf{r}_{k-1} + E_k\mathbf{r}_{k-2}\Big\}, \\ \mathbf{x}_k = A_k\Big\{B_k\mathbf{x}_{k-1} + E_k\mathbf{x}_{k-2} - \mathbf{r}_{k-1})\Big\}. \end{cases} \tag{4.10}$$

Equations (4.10) define a Lanczos-type algorithm. Now, we have to find the expressions of the coefficients $A_k$, $B_k$, and $E_k$ appearing in them. Therefore, we can write using Eq (2.136) we get

$$\begin{cases} E_k = -\frac{(\mathbf{y}_{k-1},\mathbf{r}_{k-1})}{(\mathbf{y}_{k-2},\mathbf{r}_{k-2})}, \\ B_k = \frac{-(\mathbf{y}_k,\mathbf{r}_{k-1})-E_k(\mathbf{y}_{k-1},\mathbf{r}_{k-2})}{(\mathbf{y}_{k-1},\mathbf{r}_{k-1})}, \\ A_k = \frac{1}{B_k+E_k}. \end{cases} \tag{4.11}$$

After gathering all these formulae, thus, we finally obtain the following algorithm also known as $A_4$/*Orthores* [4]

**Algorithm 12** Lanczos-type Algorithm based on relation $A_4$

**Input:** $A$ an $n \times n$ matrix, $\mathbf{b}$ an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $||\mathbf{r}_k||$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, such that $\mathbf{y} \neq 0$ and the tolerance $\varepsilon$ to $1E-13$.
Set $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$; $\mathbf{y}_0 = \mathbf{y}$;
**Compute:**
$\mathbf{r}_1$, $\mathbf{x}_1$, as in (2.138) [33];
$k = 0$;
**While** $||\mathbf{r}_k|| > \varepsilon$ **do**
$\mathbf{y}_{k+1} = A^T\mathbf{y}_k$;
$A_k$, $B_k$ and $E_k$, for $k \geq 1$, and $E_1 = 0$ as in (4.11)
$\mathbf{r}_k$ and $\mathbf{x}_k$ as in (4.10)
$k = k + 1$;
**EndWhile**
Obtain the approximate solution as well as the residual norm.
$\text{sol}_{last} = \mathbf{x}_k$;
$\text{norm}_{last} = ||\mathbf{r}_k||$;
**Stop**.

### 4.2.3 Lanczos-type Algorithm Based on Relations $A_8/B_{10}$

This kind combination is known as the **Orthomin** algorithm [4]. The algorithm $A_8/B_{10}$ is based on recurrence relations $A_8$ and $B_{10}$ [4].

#### 4.2.3.1 Formula $A_8$

The formula $A_8$ is obtained by calculating recursively the family of orthogonal polynomial $P_k$ from $P^{(1)}_{k-1}$ and $P_{k-1}$. Consider the relation below

$$P_k(x) = (A_k x + B_k)P^{(1)}_{k-1} + (C_k x + D_k)P_{k-1}. \tag{4.12}$$

The constant coefficients $A_k$, $B_k$, $C_k$, and $D_k$, are determined by the normalization condition $P_k(0) = 1$ and imposing the orthogonality condition (2.1). For the detailed derivation, the identification of the coefficients and the algorithm itself, please refer to [4].

From the above we immediately obtain

$$P_k(x) = A_k x P^{(1)}_{k-1} + P_{k-1}. \tag{4.13}$$

Their coefficients are estimated as $B_k = 0$, $C_k = 0$, $D_k = 1$, and

$$A_k = -\frac{c(x^{k-1}P_{k-1})}{c(x^k P^{(1)}_{k-1})}. \tag{4.14}$$

Since $\mathbf{r}_k = P_k(A)\mathbf{r}_0$, the equation (4.14), after replacing $x$ by $A$ and using $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, we get

$$\begin{cases} \mathbf{r}_k = \mathbf{r}_{k-1} + A_k \mathbf{z}_{k-1}, \\ \mathbf{x}_k = \mathbf{x}_{k-1} - A_k \mathbf{z}_{k-1}. \end{cases} \tag{4.15}$$

with $z_k$ defined in Eq (4.20)

Equations (4.15) define a Lanczos-type algorithm. Now, we have to find the expression of the coefficients $A_k$, appearing in them. Therefore, we can write using Eq (2.136) we get

$$A_k = -\frac{(\mathbf{y}_{k-1}, \mathbf{r}_{k-1})}{(\mathbf{y}_{k-1}, A\mathbf{z}_{k-1})}, \tag{4.16}$$

#### 4.2.3.2 Formula $B_{10}$

Consider the relation

$$P^{(1)}_k(x) = (A^1_k x + B^1_k)P^{(1)}_{k-1} + C^1_k P_k, \tag{4.17}$$

The constant coefficients $A^1_k$, $B^1_k$, and $C^1_k$, are determined by imposing the orthogonality condition (2.2). For the detailed derivation, the identification of the coefficients and the algorithm itself, please refer to [4].

From the above we immediately obtain

$$P^{(1)}_k(x) = B^1_k P^{(1)}_{k-1} + C^1_k P_k. \tag{4.18}$$

Their coefficients are estimated as $A_k^1 = 0$, with $C_k^1 = \frac{1}{a_k}$ and $a_k$ being the coefficient of $x^k$ in $P_k(x) = a_k x^k + ... + 1$, we have, $a_k = A_k C_{k-1}^1 a_{k-1} = A_k$.

$$\begin{cases} B_k^1 = -\frac{C_k^1 c(x^k P_k)}{c(x^k P_{k-1}^{(1)})}, \\ C_k^1 = \frac{1}{A_k}. \end{cases} \tag{4.19}$$

Since $\mathbf{z}_k = P_k^{(1)}(A)\mathbf{r}_0$, the equation (4.19), after replacing $x$ by $A$, we get

$$\mathbf{z}_k = B_k^1 \mathbf{z}_{k-1} + C_k^1 \mathbf{r}_k, \tag{4.20}$$

Now, we have to find the expression of the coefficients $B_k^1$, and $C_k^1$ appearing in them. Therefore, we can write using Eq (2.136) we get

$$\begin{cases} B_k^1 = -\frac{C_k^1 (\mathbf{y}_k, \mathbf{r}_k)}{(\mathbf{y}_{k-1}, A\mathbf{z}_{k-1})}, \\ C_k^1 = \frac{1}{A_k}. \end{cases} \tag{4.21}$$

Thus we finally obtain algorithm $A_8/B_{10}$ [4]

**Algorithm 13** Lanczos-type Algorithm based on relations $A_8/B_{10}$

**Input:** $A$ an $n \times n$ matrix, $\mathbf{b}$ an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $||\mathbf{r}_k||$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, such that $\mathbf{y} \neq 0$ and the tolerance $\varepsilon$ to $1E-13$.
Set $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$; $\mathbf{y}_0 = \mathbf{y}$, $\mathbf{z}_0 = \mathbf{r}_0$;
**Compute:**
$\mathbf{y}_1 = A^T \mathbf{y}_0$; $A_1$ as in (4.16);
$k = 0$;
**While** $||\mathbf{r}_k|| > \varepsilon$ **do**
$\mathbf{y}_k = A^T \mathbf{y}_{k-1}$;
$A_k$, as in (4.16);
$B_k^1$, $C_k^1$ as in (4.21);
$\mathbf{r}_k$, $\mathbf{x}_k$ as in (4.15);
$\mathbf{z}_k$ as in (4.20);
$k = k + 1$;
**EndWhile**
Obtain the approximate solution as well as the residual norm.
$\text{sol}_{last} = \mathbf{x}_k$;
$\text{norm}_{last} = ||\mathbf{r}_k||$;
**Stop**.

### 4.2.4 Lanczos-type Algorithm Based on Relations $A_8/B_6$

The implementation of this combination is known as the **Orthodir** algorithm [4]. The algorithm is based on recurrence relations $A_8$ and $B_6$ [4].

#### 4.2.4.1 Formula $B_6$

Consider the relation below

$$P_k^{(1)}(x) = (A_k^1 x^2 + B_k^1 x + C_k^1)P_{k-2}^{(1)} + (D_k^1 x + E_k^1)P_{k-1}^{(1)}. \tag{4.22}$$

The constant coefficients $A_k^1$, $B_k^1$, $C_k^1$, $D_k^1$ and $E_k^1$ are determined by imposing the orthogonality condition (2.1). For the detailed derivation, the identification of the coefficients and the algorithm itself, please refer to [4].

From the above we immediately obtain

$$P_k^{(1)}(x) = C_k^1 P_{k-2}^{(1)} + (x + E_k^1)P_{k-1}^{(1)}, \tag{4.23}$$

Their coefficients are estimated as $A_k^1 = 0$, $B_k = 0$, $D_k^1 = 1$ and

$$\begin{cases} C_k^1 = -\frac{c(x^k P_{k-1}^{(1)})}{c(x^{k-1} P_{k-2}^{(1)})}, \\ E_k^1 = -\frac{-c(x^{k+1} P_{k-1}^{(1)}) - C_k^1 c(x^k P_{k-2}^{(1)})}{c(x^k P_{k-1}^{(1)})}. \end{cases} \tag{4.24}$$

Since $\mathbf{z}_k = P_k^{(1)}(A)\mathbf{r}_0$, the equation (4.23), after replacing $x$ by $A$, we get

$$\mathbf{z}_k = C_k^1 \mathbf{z}_{k-2} + E_k^1 \mathbf{z}_{k-1} + A\mathbf{z}_{k-1}, \tag{4.25}$$

Now, we have to find the expression of the coefficients $C_k^1$, and $E_k^1$ appearing in them. Therefore, we can write using Eq (2.136) we get with

$$\begin{cases} C_k^1 = -\frac{(\mathbf{y}_k, \mathbf{z}_{k-1})}{(\mathbf{y}_{k-1}, \mathbf{z}_{k-2})}, \\ E_k^1 = -\frac{-(\mathbf{y}_k, A\mathbf{z}_{k-1}) - C_k^1 (\mathbf{y}_k, \mathbf{z}_{k-2})}{(\mathbf{y}_k, \mathbf{z}_{k-1})}. \end{cases} \tag{4.26}$$

Let us now design an algorithm which combines $A_8$ and $B_6$ for the computation of the residuals $\mathbf{r}_k$, the corresponding vectors $\mathbf{x}_k$ from $A_8$ of section 4.2.3.1, and $\mathbf{z}_k$, from $B_6$ of section 4.2.4.1. Thus we finally obtain the following algorithm $A_8/B_6$ [4].

**Algorithm 14** Lanczos-type Algorithm based on relations $A_8/B_6$

**Input:** $A$ an $n \times n$ matrix, $\mathbf{b}$ an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $||\mathbf{r}_k||$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, and the tolerance $\varepsilon$ to $1.0E-13$.
Set $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$; $\mathbf{y}_0 = \mathbf{y}$, $\mathbf{z}_0 = \mathbf{r}_0$;
**Compute:**
$\mathbf{r}_1$, $\mathbf{x}_1$, as in (2.138), $\mathbf{z}_1$ as in (2.146);
$k = 0$;
**While** $||\mathbf{r}_k|| > \varepsilon$ **do**
$\mathbf{y}_k = A^T\mathbf{y}_{k-1}$;
$A_k$, as in (4.16) and $C^1_{k+1}, E^1_{k+1}$ as in (4.26) respectively;
$\mathbf{r}_k$, $\mathbf{x}_k$ as in (4.15) and $\mathbf{z}_k$ as in (4.25) respectively;
$k = k + 1$;
**EndWhile**
Obtain the approximate solution as well as the residual norm.
$\text{sol}_{last} = \mathbf{x}_k$;
$\text{norm}_{last} = ||\mathbf{r}_k||$;
**Stop**.

## 4.3 Numerical Results

The experimental results which are recorded in Table 4.1 show that algorithms $A_4$, $A_{12}$, $A_8/B_6$ and $A_8/B_{10}$ solved the problem up to dimension 20. These algorithms failed for $n \geq 30$ and above. The reason is that the Lanczos-type algorithms breaks down. Since all algorithms of this type are based on recurrence relationships between FOPs $P_k(x)$ and $P^{(1)}_k(x)$, the polynomials involve the computation of some scalar products appearing as denominators and numerators of the coefficients of the recursive relationships. Some of the denominators becomes smaller than $1.0E-14$ which causes breakdown in these algorithms and they have to be stopped. The breakdown is also due to the non-existence of some polynomials $P_k(x)$. This breakdown issue will be discussed and addressed in Section 4.4.

**Table 4.1:** *Results of Lanczos-type algorithms on Baheux-type problems for* $\delta = 0$

| Dim of Prob | $A_4$ | | $A_{12}$ | | $A_8/B_6$ | | $A_8/B_{10}$ | |
|---|---|---|---|---|---|---|---|---|
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\|\| r_k \|\|$ | t(sec) | $\|\| r_k \|\|$ | t(sec) |
| 10 | 3.7525E-14 | 9.0644E-01 | 2.2493E-14 | 8.0559E-01 | 6.4731E-16 | 9.0150E-01 | 3.8369E-14 | 8.0912E-01 |
| 20 | 5.2880E-14 | 1.0880E+00 | 8.6013E-14 | 1.4494E+00 | 4.2156E-14 | 8.7512E-01 | 1.4607E-14 | 9.5158E-01 |
| 30 | NaN | | NaN | | NaN | | NaN | |
| 100 | NaN | | NaN | | NaN | | NaN | |

## 4.4 Pre-emptive restarting approach to Lanczos-type algorithms

The causes of breakdown in the most common Lanczos-type algorithms can be found by monitoring the components of the coefficients that blow up prior to breakdown. Our aim is to investigate the behaviour of the coefficients involved in the recurrence relations and the parameters of the offending coefficents/denominators of the Lanczos algorithm under consideration. When any of these offending denominators/coefficents goes to zero/NaN the Lanczos algorithm fails. The NaN situation arises due to overflow or underflow of the coefficients involved [39, 46, 47]. After careful monitoring, the coefficients which cause the breakdown will be identified. A possible remedy to avoid this problem could be to design a test/rule by which the Lanczos algorithm can be stopped before breakdown. This is referred to as the break statement. The test might be based on choosing a threshold value $\epsilon$, for instance, for that parameter in the coefficients which caused breakdown. After deciding on the threshold, restarting/switching with a pre-emption approach can be implemented.

**Algorithm 15** Monitoring Lanczos-type Algorithms

**Description:**

1: Choose Lanczos-type algorithms based on $\{A_4, A_{12}, A_8/B_6, A_8/B_{10}\}$
2: Monitor coefficients and denominators:
3: Design a test/rule. The test might be based on choosing a threshold value $\epsilon$, for instance, for that parameter in the coefficients which caused breakdown:
4: Obtain the approximate solution as well as the residual norm.
5: Stop.

### 4.4.1 Monitoring Lanczos-type Algorithm based on relation $A_{12}$

As an example, the behaviour of coefficients used in Algorithm $A_{12}$ has been investigated for $\delta = 0, 0.2, 5$ and 8. First, consider the case of $\delta = 0$. It can be seen in Table 4.2 that the problem of breakdown is caused by the coefficient $A_{k+1}$ whose values for various dimensions are given in column 8 of the table. The corresponding dimensions are given in the first column of the table and range from 100 to 90000. The coefficient values in column 8 are actually the additive combination of columns 5 and 7. Both column 5 and column 7 seem to have blown up (showing $NaN$) when column 8 is $NaN$. Therefore, it is important to concentrate on each of column 5 and column 7 to see which of their building component is the culprit. To this end, all the coefficients $A_k$, $B_k$, $C_k$, $F_k$, $G_k$ and $\Delta_k$, can be written in terms of $a_{ij}, i = 1, 2, 3; j = 1, 2, 3$ and see which of them causes the breakdown. There will be a compound term in the expression of the coefficients or cluster of $a_{ij}$ which blows up (i.e. goes to NaN or $\infty$). While monitoring the $A_{12}$ algorithm, it turns out that breakdown is caused by $a_{11}$ and $a_{13}$. As shown in Table 4.3, the behaviour of these coefficients is monitored by trying various values starting from the highest possible value of $1.0E + 103$ and $1.0E + 102$ for $a_{11}$ and $a_{13}$, respectively, and reach the final default values at which the algorithm does not breakdown for the size of the problem ranging from 100 to 90000. The observed default values are $1.0E + 80$ and $1.0E + 80$ for $a_{11}$ and $a_{13}$, respectively.

As has been mentioned above, the $A_{12}$ algorithm is also investigated for systems generated through discretisation of an integral operator for values $\delta$= 0.2, 5 and 8 as done for $\delta = 0$, for $\delta = 0.2$, shown in Table A.1. The behaviour of the culprit coefficients is monitored by trying various values starting from the highest possible value of $1.0E+103$ and $1.0E+102$

for $a_{11}$ and $a_{13}$, respectively as shown in Table A.4. The final default values for which the algorithm does not breakdown are ultimately reached. The observed default values are $1.0E+90$ and $1.0E+90$ for $a_{11}$ and $a_{13}$, respectively. For $\delta = 5$ as shown in Table A.2, the observed default values are $1.0E+90$ and $1.0E+90$ for $a_{11}$ and $a_{13}$, respectively, with the starting highest values the same as those for for $\delta = 0.2$ as shown in Table A.5. Similarly, for $\delta = 8$ the behaviour of coefficients as shown in Table A.3, and the observed default values are $1.0E+95$ and $1.0E+95$ for $a_{11}$ and $a_{13}$, respectively, with the starting highest possible values of $1.000E+104$ and $1.000E+101$ for $a_{11}$ and $a_{13}$, respectively, as shown in Table A.6. The numerical evidence for the above scenario are recorded in Tables 4.2-A.6. Similar tables

**Table 4.2:** *Behaviour of coefficients of $A_{12}$ on Baheux-type problems when $\delta = 0$.*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 | Col.6 | Col.7 | Col.8 |
|---|---|---|---|---|---|---|---|
| Dim. of $A$ | $k$ | $\Delta_k$ | $B_k$ | $C_k$ | $F_k$ | $G_k$ | $A_k$ |
| 100 | 148 | Inf | NaN | NaN | -4.8175E-01 | NaN | NaN |
| 500 | 140 | Inf | NaN | NaN | -2.1018E+01 | NaN | NaN |
| 1000 | 138 | -1.6699E+307 | -Inf | NaN | -2.3393E+01 | Inf | NaN |
| 5000 | 139 | 6.9121E+307 | NaN | NaN | 1.1891E+01 | NaN | NaN |
| 10000 | 139 | NaN | NaN | NaN | 7.9101E+00 | NaN | NaN |
| 15000 | 137 | -4.9561E+303 | 7.6200E+00 | Inf | 0.0000E+00 | 7.4146E+00 | 0.0000E+00 |
| 20000 | 135 | -1.5143E+306 | -Inf | Inf | 5.4000E+01 | -Inf | NaN |
| 30000 | 138 | Inf | NaN | NaN | 3.1908E+00 | NaN | NaN |
| 40000 | 139 | NaN | NaN | NaN | -5.6457E+02 | NaN | NaN |
| 50000 | 122 | 3.3211E+263 | 1.1387E+02 | NaN | 0.0000E+00 | 8.0000E+01 | NaN |
| 60000 | 138 | -Inf | NaN | NaN | 1.2075E-01 | NaN | NaN |
| 70000 | 138 | -Inf | NaN | NaN | 4.7600E+02 | NaN | NaN |
| 80000 | 139 | Inf | NaN | NaN | 7.8443E+00 | NaN | NaN |
| 90000 | 139 | 1.1064e+308 | -Inf | NaN | 3.0815E+00 | -Inf | NaN |

are generated for different instances of the problem. These can be seen as Tables A.1-A.3, subsection A.2.1 of Appendix A. The purpose of these tables is to show that monitoring by coefficients helps to avoid breakdown. As these tables show, as soon as any of the entries in a row hits infinity or is Not a Number (Inf or NaN), the Lanczos-type algorithm breaks down. Note that, while Table 4.2 shows the values of compound coefficient such

**Table 4.3:** *Behaviour of the parameters of the offending coefficients of $A_{12}$ on Baheux-type problems when* $\delta = 0$.

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 | Col.6 | Col.7 | Col.8 |
|---|---|---|---|---|---|---|---|
| Dim. of $A$ | $k$ | $a_{11}$ | $a_{13}$ | $a_{21}$ | $a_{23}$ | $a_{31}$ | $a_{33}$ |
| 100 | 148 | 1.1549E+102 | 2.3972E+102 | 2.7997E+101 | 3.7936E+103 | -3.3596E+102 | 1.3998E+104 |
| 500 | 140 | 5.9603E+101 | 2.8357E+100 | 1.1013E+104 | 1.1516E+102 | 1.7730E+105 | 5.5652E+103 |
| 1000 | 138 | 1.8548E+102 | 7.9288E+100 | 3.5810E+103 | 1.5792E+102 | 6.1768E+104 | 2.2485E+103 |
| 5000 | 139 | -2.6553E+102 | 2.2330E+101 | -5.5792E+103 | 3.5762E+102 | -9.2291E+104 | 6.3973E+103 |
| 10000 | 139 | -6.1593E+102 | 7.7866E+101 | -2.1502E+104 | 1.1339E+103 | -3.8344E+105 | 1.8982E+104 |
| 15000 | 137 | 0.0000E+00 | -4.4839E+100 | 3.3246E+101 | -6.3430E+101 | 2.1698E+102 | -9.4839E+102 |
| 20000 | 135 | -2.9528E+101 | 5.4681E+99 | -7.1917E+102 | -1.6536E+102 | -1.4320E+104 | -5.8233E+103 |
| 30000 | 138 | 1.8285E+102 | -5.7306E+101 | 3.9895E+103 | -1.0271E+103 | 7.5983E+104 | -1.5482E+104 |
| 40000 | 139 | -6.9152E+103 | -1.2249E+101 | -2.1726E+105 | -3.2196E+102 | -4.2609E+106 | -1.4558E+103 |
| 50000 | 122 | 0.0000E+00 | 3.7299E+86 | -2.9839E+88 | 3.7797E+88 | 3.7399E+89 | 8.0765E+89 |
| 60000 | 138 | 3.4996E+100 | -2.8981E+101 | -2.9257E+103 | -1.2144E+103 | -8.0295E+104 | -2.7647E+104 |
| 70000 | 138 | -1.2494E+103 | 2.6247E+100 | -2.2341E+104 | 1.2074E+102 | -3.5254E+105 | 2.0158E+103 |
| 80000 | 139 | -3.5258E+102 | 4.4948E+101 | -6.4533E+103 | 1.5258E+103 | -1.1305E+105 | 3.8209E+104 |
| 90000 | 139 | -1.8198E+102 | 5.9056E+101 | -2.6457E+103 | 8.2591E+102 | -3.2924E+104 | 1.3550E+104 |

as $\Delta_k$, $A_k$, $B_k$, $C_k$, $F_k$ and $G_k$, Table 4.3 involves the particular parameters of the compound coefficient which is responsible for the breakdown. Therefore, it is potentially cheaper to check for breakdown in Table 4.3 than in Table 4.2. Similar tables for different instance can be found in Tables A.4-A.6, Subsection A.2.1 of Appendix A. After gathering all these, thus, we finally obtain the following algorithm

**Algorithm 16** Monitoring Lanczos-type Algorithms based on relation $A_{12}$

**Input:** $A$ an $n \times n$ matrix, $\mathbf{b}$ an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $\|\mathbf{r}_k\|$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, such that $y \neq 0$ and the tolerance $\varepsilon$ to $1E-13$.
Set $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$; $\mathbf{y}_0 = \mathbf{y}$;
**Compute:**
$\mathbf{r}_1$, $\mathbf{x}_1$, $\mathbf{r}_2$ and $\mathbf{x}_2$ as in (2.138) and (2.139), [33]
$k = 2$;
**While** $\|\mathbf{r}_k\| > \varepsilon$
$\mathbf{y}_{k+1} = A^T\mathbf{y}_k$;
$B_k$ as in (4.3)
$A_k$, $C_k$, and $G_k$, as in (4.4),
$F_k$ as in (4.6);
$\mathbf{r}_k$ and $\mathbf{x}_k$ as in (4.5)
/* Monitor coefficients and denominators: $A_k$, $B_k$, $C_k$, $F_k$, $G_k$, $a_{11}$, $a_{13}$.*/
/* Design a test/rule. The test might be based on choosing a threshold value $\epsilon$, for instance, for that parameter in the coefficients which caused breakdown. */

**Algorithm 16** Lanczos-type Algorithm based on relations $A_{12}$ (continued)

**If** ($|a_{11}| \leq 1.0E-25$);
  display('Check zero ......');
  **break**;
**End**;
**If** ( $|a_{11}| \geq \omega_i$ and $|a_{13}| \geq \omega_i$ )
  display('Check Yes ......');
  **break**;
**End**;

where $\omega_i = 1.0E+80, 1.0E+90$, $1.0E+90$, $1.0E+95$
for different $\delta_i = 0$, 0.2, 5, 8, when $i = 1, 2, 3, 4$ respectively;
$k = k + 1$;
**EndWhile**
Obtain the approximate solution as well as the residual norm;
$\text{sol}_{last} = \mathbf{x}_k$;
$\text{norm}_{last} = \|\mathbf{r}_k\|$;
**Stop**.

### 4.4.2 Monitoring Lanczos-type Algorithm based on relation $A_4$ (Orthores)

Similarly to monitoring $A_{12}$, the behaviour of coefficients used in Algorithm $A_4$ has also been investigated for $\delta$= 0, 0.2, 5 and 8. Here also, the behaviour of coefficients for $\delta = 0$ are considered first. It can be seen in Table 4.4 that the problem of breakdown is caused by the coefficient $B_{k+1}$ whose values for various dimensions of the test problems are given in column 4 of the table. The corresponding dimensions are given in the first column of the table that range from 100 to 90000. The coefficient values in column 4 seem to have blown up showing $\pm\infty$ or $NaN$. Therefore it becomes important to concentrate on $B_k$ as a good term to observe in order to detect breakdown. Moreover, when $B_k$ takes $\pm\infty$, $a_k$ is always 0. Therefore, one can design a test in two parts, one on $a_k$ and the other on $c_k$. To this end, all the coefficients $A_k$, $B_k$ and $E_k$ can be written in terms of $a_k$, $b_k$, $c_k$ and $d_k$ to see which cluster of these causes the breakdown. Like in $A_{12}$ algorithm, there will be a compound term in the expression of the coefficients or cluster of these which blows up (i.e. goes to NaN or ∞). The components that cause the breakdown are $a_k$ and $c_k$. As shown in Table 4.5, the behaviour of these coefficients is monitored by trying various values starting from the highest possible value of $1.0E + 292$ and $1.0E + 287$ for $a_k$ and $c_k$, respectively, and reach the final default values at which the algorithm does not breakdown for the size of the problem ranging from 100 to 90000. The observed default values are $1.0E + 124$ and $1.0E + 125$ for $a_k$ and $c_k$, respectively.

Furthermore, the $A_4$ algorithm is also investigated for discretisation values $\delta$= 0.2, 5 and 8. Similar to $\delta = 0$, for $\delta = 0.2$, as shown in Table A.7 the behaviour of the culprit coefficients is monitored by trying various values starting from the highest possible value of $1.0E + 291$ and $1.0E + 262$ for $a_k$ and $c_k$, as shown in Table A.10, respectively, and reaching the final default values at which the algorithm does not breakdown for the size of the problem ranging from 100 to 90000. The observed default values are $1.0E + 118$ and $1.0E + 119$ for $a_k$ and $c_k$, respectively. In a similar fashion, the behaviour of the coefficients for the value of $\delta = 5$ are shown in Table A.8. The observed default values are $1.0E + 275$ and $1.0E + 277$ for $a_k$ and $c_k$, respectively, with the starting highest values being $1.0E + 293$ and $1.0E + 291$ as show in Table A.11. Similarly, for $\delta = 8$ the observed default values are $1.0E + 277$ and $1.0E + 278$ for $a_k$ and $c_k$, respectively, with the starting highest values are $1.0E + 295$ and $1.0E + 295$ as shown in Table A.12. The numerical evidence for the above scenario relating to algorithm $A_4$ are recorded in Tables 4.4-A.12

Similar tables are generated for different instances of the problem. These can be seen as Tables A.7-A.9, subsection A.2.2 of Appendix A. The purpose of these tables is to show

**Table 4.4:** *Behaviour of coefficients of $A_4$ on Baheux-type problems when $\delta = 0$.*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $A_k$ | $B_k$ | $E_k$ |
| 100 | 358 | NaN | NaN | -3.1373E+00 |
| 500 | 352 | NaN | NaN | 1.8452E+01 |
| 1000 | 352 | NaN | NaN | 4.6160E-01 |
| 5000 | 352 | NaN | NaN | 6.9004E+02 |
| 10000 | 182 | 0.0000E+00 | Inf | 0.0000E+00 |
| 15000 | 257 | 0.0000E+00 | -Inf | 0.0000E+00 |
| 20000 | 237 | 0.0000E+00 | -Inf | 0.0000E+00 |
| 30000 | 311 | NaN | NaN | 0.0000E+00 |
| 40000 | 319 | 0.0000E+00 | Inf | 0.0000E+00 |
| 50000 | 227 | 0.0000E+00 | -Inf | 0.0000E+00 |
| 60000 | 352 | NaN | NaN | 1.4815E+01 |
| 70000 | 147 | 0.0000E+00 | Inf | 0.0000E+00 |
| 80000 | 345 | 0.0000E+00 | Inf | 0.0000E+00 |
| 90000 | 352 | NaN | NaN | 1.6602E+01 |

that monitoring by coefficients helps to avoid breakdown. As these tables show, as soon as any of the entries in a row hits infinity or is Not a Number (Inf or NaN), the Lanczos-type algorithm breaks down. Note that, while Table 4.4 shows the values of compound

**Table 4.5:** *Behaviour of the parameters of the offending coefficients of $A_4$ on Baheux-type problems when $\delta = 0$*

| *Col.1* | *Col.2* | *Col.3* | *Col.4* | *Col.5* | *Col.6* |
|---|---|---|---|---|---|
| *Dim. of A* | $k$ | $a_k$ | $b_k$ | $c_k$ | $d_k$ |
| 100 | 358 | 2.2305E+285 | 7.1098E+284 | NaN | 3.9406E+285 |
| 500 | 352 | 2.2297E+292 | -1.2084E+291 | NaN | 3.4070E+292 |
| 1000 | 352 | -8.6946E+289 | 1.8836E+290 | NaN | 1.8598E+291 |
| 5000 | 352 | -5.5848E+291 | 8.0935E+288 | NaN | 1.1405E+290 |
| 10000 | 182 | 0.0000E+00 | -2.1331E+141 | -3.2391E+141 | -3.5654E+142 |
| 15000 | 257 | 0.0000E+00 | 8.5280E+205 | 8.8290E+206 | 1.3845E+207 |
| 20000 | 237 | 0.0000E+00 | -1.4141E+187 | 2.0885E+188 | 4.7862E+188 |
| 30000 | 311 | 0.0000E+00 | 1.1914E+253 | 0.0000E+00 | -1.4663E+253 |
| 40000 | 319 | 0.0000E+00 | -3.5363E+260 | -1.3530E+262 | -1.7835E+261 |
| 50000 | 227 | 0.0000E+00 | -1.0455E+180 | 2.3341E+180 | -1.3162E+181 |
| 60000 | 352 | 1.2182E+290 | -8.2226E+288 | NaN | 4.3854E+289 |
| 70000 | 147 | 0.0000E+00 | -3.5815E+109 | -8.8305E+110 | -9.4646E+110 |
| 80000 | 345 | 0.0000E+00 | -5.4950E+284 | -1.9034E+287 | -5.9202E+285 |
| 90000 | 352 | 1.6811E+290 | -1.0126E+289 | NaN | 2.0709E+290 |

coefficient such as $A_k$, $B_k$, and $E_k$, Table 4.5 involves the particular parameters of the compound coefficient which is responsible for the breakdown. Therefore, it is potentially cheaper to check for breakdown in Table 4.5 than in Table 4.4. Similar tables for different instance can be found in Tables A.10-A.12, Subsection A.2.2 of Appendix A. After gathering all these, thus, we finally obtain the following algorithm

**Algorithm 17** Monitoring Lanczos-type Algorithm based on relation $A_4$

**Input:** $A$ an $n \times n$ matrix, $\mathbf{b}$ an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $\|\mathbf{r}_k\|$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, and the tolerance $\varepsilon$ to $1E-13$.
Set $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$; $\mathbf{y}_0 = \mathbf{y}$;
**Compute:**
$\mathbf{r}_1$, $\mathbf{x}_1$, as in (2.138) [33];
$k = 0$;
**While** $\|\mathbf{r}_k\| > \varepsilon$
$\mathbf{y}_k = A^T\mathbf{y}_{k-1}$;
$A_k$, $B_k$ and $E_k$, for $k \geq 1$, and $E_1 = 0$ as in (4.11);
$\mathbf{r}_k$ and $\mathbf{x}_k$ as in (4.10)
/* Monitor Denominators: $A_k$, $B_k$, $E_k$, $a_k$, $c_k$.*/
/* Design a test/rule. The test might be based on choosing a threshold value $\epsilon$, for instance, for that parameter in the coefficients which caused breakdown. */

**If** ($|a_k| \leq 1.0E-25$);
display('Check zero ......');
**break**;
**End**;
**If** ($|a_k| \geq \alpha_i$ and $|c_k| \geq \beta_i$)
display('Check Yes ......');
**break**;
**End**;

where $\alpha_i = 1.0E+124, 1.0E+118, 1.0E+275, 1.0E+277$;
where $\beta_i = 1.0E+125, 1.0E+119, 1.0E+277, 1.0E+278$;
when $i = 1, 2, 3, 4$, for different $\delta_i = 0, 0.2, 5, 8$; respectively;
$k = k+1$;
**EndWhile**
Obtain the approximate solution as well as the residual norm;
$\text{sol}_{last} = \mathbf{x}_k$;
$\text{norm}_{last} = \|\mathbf{r}_k\|$;
**Stop**.

### 4.4.3 Monitoring Lanczos-type Algorithm based on relations $A_8/B_6$

Here also, the behaviour of the coefficients used in Algorithm $A_8/B_6$ have been investigated for $\delta$= 0, 0.2, 5 and 8. The behaviour for $\delta = 0$ is considered first. It can be seen in Table 4.6 that the problem of breakdown is caused by the coefficient $E_k^1$ whose values for various dimension are given in column 5 of the table. The corresponding dimensions are given in the first column of the table that range from 100 to 90000. The coefficient values in column 5 seem to have blown up showing *NaN*. Therefore it becomes important to concentrate on $E_k^1$ as a good term to observe in order to detect breakdown. To this end, all the coefficients $A_k$, $C_k^1$ and $E_k^1$, can be written in terms of $a_k$, $b_k$ and $c_k$ to see which cluster of these causes the breakdown. Like in $A_4$ and $A_{12}$ algorithms, there is a term in the expression of the coefficients which blows up (i.e. goes to NaN or zero). The components that cause the breakdown are $b_k$. As shown in Table 4.7, the behaviour of these coefficients is monitored by trying various values starting from the highest possible value of $1.0E + 295$ for $b_k$, and reach the final default values at which the algorithm does not breakdown for the size of the problem ranging from 100 to 90000. The observed default values are $1.0E + 90$ for $b_k$. Furthermore, the $A_8/B_6$ algorithm is also investigated for discretisation values $\delta$= 0.2, 5 and 8. Similarly to $\delta = 0$, for $\delta = 0.2$, as shown in Table A.13, the behaviour of the culprit coefficients is monitored by trying various values starting from the highest possible value of $1.0E + 294$ for $b_k$, as shown in Table A.16, and reaching the final default values at which the algorithm does not breakdown for the size of the problem ranging from 100 to 90000. The observed default values are $1.0E + 130$ for $b_k$. In a similar fashion, the behaviour of the coefficients is monitored for the value of $\delta = 5$ shown in Table A.14. The observed default values are $1.0E+280$ for $b_k$, with the starting highest value $1.0E+294$ as show in Table A.17. Similarly, the behaviour of coefficients for $\delta = 8$ are shown in Table A.15. The observed default values are $1.0E + 290$ for $b_k$, with the starting highest value being $1.0E + 294$ as shown in Table A.18. The numerical evidence for the above scenario relating to algorithm $A_8/B_6$ are recorded in Tables 4.6-A.18

**Table 4.6:** *Behaviour of coefficients of $A_8/B_6$ on Baheux-type problems when $\delta = 0$.*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $A_k$ | $C_k^1$ | $E_k^1$ |
| 100 | 45 | NaN | NaN | NaN |
| 500 | 176 | NaN | NaN | NaN |
| 1000 | 174 | NaN | NaN | NaN |
| 5000 | 174 | NaN | NaN | NaN |
| 10000 | 176 | NaN | NaN | NaN |
| 15000 | 174 | NaN | NaN | NaN |
| 20000 | 174 | NaN | NaN | NaN |
| 30000 | 168 | NaN | NaN | NaN |
| 40000 | 174 | NaN | NaN | NaN |
| 50000 | 133 | NaN | NaN | NaN |
| 60000 | 129 | NaN | NaN | NaN |
| 70000 | 171 | NaN | NaN | NaN |
| 80000 | 170 | NaN | NaN | NaN |
| 90000 | 177 | NaN | NaN | NaN |

Similar tables are generated for different instances of the problem. These can be seen as Tables A.13-A.15, subsection A.2.3 of Appendix A. The purpose of these tables is to show that monitoring by coefficients helps to avoid breakdown. As these tables show, as soon as any of the entries in a row hits infinity or is Not a Number (Inf or NaN), the Lanczos-type algorithm breaks down.

Note that, while Table 4.6 shows the values of compound coefficient such as $A_k$, $C_k^1$, and $E_k^1$, Table 4.7 involves the particular parameters of the compound coefficient which is responsible for the breakdown. Therefore, it is potentially cheaper to check for breakdown

**Table 4.7:** *Behaviour of the parameters of the offending coefficients of* $A_8/B_6$ *on Baheux-type problems when* $\delta = 0$

| *Col.1* | *Col.2* | *Col.3* | *Col.4* | *Col.5* | *Col.6* | *Col.7* |
|---|---|---|---|---|---|---|
| *Dim. of A* | $k$ | $a_k$ | $b_k$ | $c_k$ | $f_k$ | $e_k$ |
| 100 | 45 | 5.0320E+13 | NaN | -4.3429E+40 | -4.4917E+41 | NaN |
| 500 | 176 | 2.1279E+144 | NaN | 2.2442E+292 | 1.1997E+293 | NaN |
| 1000 | 174 | 7.0826E+141 | NaN | NaN | -3.7561E+294 | NaN |
| 5000 | 174 | 2.7973E+142 | NaN | NaN | 1.7832E+294 | NaN |
| 10000 | 176 | -2.0730E+143 | NaN | NaN | NaN | NaN |
| 15000 | 174 | 7.7219E+138 | NaN | NaN | -5.9177E+294 | NaN |
| 20000 | 174 | -8.2407E+141 | NaN | NaN | NaN | NaN |
| 30000 | 168 | 1.8353E+136 | NaN | NaN | 1.9360E+294 | NaN |
| 40000 | 174 | 3.4297E+142 | NaN | -5.4886E+292 | -4.3877E+293 | NaN |
| 50000 | 133 | -7.7943E+104 | NaN | 3.5244E+222 | 2.2999E+223 | NaN |
| 60000 | 129 | NaN | NaN | -4.0132E+205 | 5.1369E+207 | NaN |
| 70000 | 171 | -2.2328E+140 | NaN | NaN | NaN | NaN |
| 80000 | 170 | 6.0473E+137 | NaN | NaN | -7.5842E+293 | NaN |
| 90000 | 177 | -6.2923E+144 | NaN | NaN | NaN | NaN |

in Table 4.7 than in Table 4.6. Similar tables for different instance can be found in Tables A.16-A.18, subsection A.2.3 of Appendix A. After gathering all these, thus, we finally obtain the following algorithm

**Algorithm 18** Monitoring Lanczos-type Algorithm based on relation $A_8/B_6$

**Input:** $A$ an $n \times n$ matrix, $b$ an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $\|\mathbf{r}_k\|$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, and the tolerance $\varepsilon$ to $1E-13$.
Set $\mathbf{r}_0 = b - A\mathbf{x}_0$; $\mathbf{y}_0 = \mathbf{y}$, $\mathbf{z}_0 = \mathbf{r}_0$;
**Compute:**
$\mathbf{r}_1$, $\mathbf{x}_1$, as in (2.138);
$\mathbf{z}_1$ as in (2.146);
$k = 0$;
**While** $\|\mathbf{r}_k\| > \varepsilon$
$\mathbf{y}_k = A^T\mathbf{y}_{k-1}$
$A_k$, as in (4.16);
$\mathbf{r}_k$, $\mathbf{x}_k$ as in (4.15);

**Algorithm 18** $A_8/B_6$ based algorithm(continued)

$C_k^1, E_k^1$ as in (4.26);
$z_k$ as in (4.25);
/* Monitor Denominators: $A_k, C_k^1, E_k^1, b_k, c_k$. */
/* Design a test/rule. The test might be based on choosing a threshold value $\epsilon$, for instance, for that parameter in the coefficients which caused breakdown. */

**If** ($|b_k| \leq 1.0E-25$ or $|c_k| \leq 1.0E-25$);
display('Check zero ......');
**break**;
**End**;
**If** ($|b_k| \geq \alpha_i$)
display('Check Yes ......');
**break**;
**End**;

where $\alpha_i = 1.0E+90,\ 1.0E+130$ when $i = 1, 2$, for $\delta_i = 0,\ 0.2$ respectively;

**If** ($|b_k| \geq \beta_i$)
display('Check Yes ......');
**break**;
**End**;

where $\beta_i = 1.0E+280, 1.0E+290$, when $i = 1, 2$, for $\delta_i = 5,\ 8$ respectively;
$k = k + 1$;
**EndWhile**
Obtain the approximate solution as well as the residual norm;
$\text{sol}_{last} = \mathbf{x}_k$;
$\text{norm}_{last} = ||\mathbf{r}_k||$;
**Stop**.

### 4.4.4 Monitoring Lanczos-type Algorithm based on relations $A_8/B_{10}$

Here also, the behaviour of coefficients for $\delta = 0$ is considered first. It can be seen in Table 4.8 that the problem of breakdown is caused by the coefficient $B_k^1$ whose values for various dimension are given in column 5 of the table. The corresponding dimensions are given in the first column of the table that range from 100 to 90000. The coefficient values in column

5 seem to have blown up showing *NaN*. Therefore, we should concentrate on $B_k^1$ as a good term to observe in order to detect breakdown. To this end, all the coefficients $A_k$, $C_k^1$ and $B_k^1$ can be written in terms of $a_k$, $b_k$ and $c_k$ to see which cluster of these causes the breakdown. Like $A_8/B_6$ algorithm, there will be a term in the expression of the coefficients which blows up (i.e. goes to NaN). The components that cause the breakdown are $b_k$. As shown in Table 4.9, the behaviour of these coefficients is monitored by trying various values starting from the highest possible value of $1.0000E+292$ for $b_k$, and reach the final default values at which the algorithm does not breakdown for the size of the problem ranging from 100 to 90000. The observed default values are $1.0000E + 130$ for $b_k$.

Furthermore, the $A_8/B_{10}$ algorithm is also investigated for discretisation values $\delta$= 0.2, 5 and 8. Similarly to $\delta = 0$, for $\delta = 0.2$, and as shown in Table A.19, the behaviour of the culprit coefficients is monitored by trying various values starting from the highest possible value of $1.000E + 293$ for $b_k$, as shown in Table A.22, and reach the final default values at which the algorithm does not breakdown for the size of the problem ranging from 100 to 90000. The observed default values are $1.0000E + 150$ for $b_k$. In a similar fashion, the behaviour of the coefficients for the value of $\delta = 5$ is shown in Table A.20. The observed default values are $1.0000E + 280$ for $b_k$, with the starting highest values $1.000E + 295$ as show in Table A.23. Similarly, the behaviour of coefficients for $\delta = 8$ is shown in Table A.21. The observed default values being $1.000E+270$ for $b_k$, with the starting highest values are $1.000E + 296$ as shown in Table A.24. The numerical evidence for the above scenario relating to algorithm $A_8/B_{10}$ are recorded in Tables 4.8-A.24.

Similar tables are generated for different instances of the problem. These can be seen as Tables A.19-A.21, subsection A.2.4 of Appendix A. The purpose of these tables is to show

**Table 4.8:** *Behaviour of the parameters of the offending coefficients of $A_8/B_{10}$ on Baheux-type problems when* $\delta = 0$

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $A_k$ | $C_k^1$ | $B_k^1$ |
| 100 | 171 | NaN | NaN | NaN |
| 500 | 182 | NaN | NaN | NaN |
| 1000 | 184 | NaN | NaN | NaN |
| 5000 | 183 | NaN | NaN | NaN |
| 10000 | 183 | NaN | NaN | NaN |
| 15000 | 143 | Inf | 0.0000E+00 | NaN |
| 20000 | 117 | NaN | NaN | NaN |
| 30000 | 124 | -Inf | 0.0000E+00 | NaN |
| 40000 | 183 | -Inf | 0.0000E+00 | NaN |
| 50000 | 184 | NaN | NaN | NaN |
| 60000 | 180 | NaN | NaN | NaN |
| 70000 | 184 | NaN | NaN | NaN |
| 80000 | 183 | NaN | NaN | NaN |
| 90000 | 177 | NaN | NaN | NaN |

that monitoring by coefficients helps to avoid breakdown. As these tables show, as soon as any of the entries in a row hits infinity or is Not a Number (Inf or NaN), the Lanczos-type algorithm breaks down.

**Table 4.9:** *Behaviour of the parameters of the offending coefficients of $A_8/B_{10}$ on Baheux-type problems when* $\delta = 0$

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $a_k$ | $b_k$ | $c_k$ |
| 100 | 171 | -2.9969E+148 | NaN | NaN |
| 500 | 182 | -9.5213E+146 | NaN | NaN |
| 1000 | 184 | -8.0820E+142 | NaN | NaN |
| 5000 | 183 | 1.2419E+144 | NaN | NaN |
| 10000 | 183 | 2.4934E+149 | NaN | NaN |
| 15000 | 143 | -9.2885E+111 | 0.0000E+00 | NaN |
| 20000 | 117 | -1.0017E+84 | NaN | NaN |
| 30000 | 124 | 7.7153E+91 | 0.0000E+00 | NaN |
| 40000 | 183 | 1.1238E+146 | 0.0000E+00 | NaN |
| 50000 | 184 | -1.5609E+146 | NaN | NaN |
| 60000 | 180 | 4.7634E+139 | NaN | NaN |
| 70000 | 184 | -2.8970E+147 | NaN | NaN |
| 80000 | 183 | -1.5784E+148 | NaN | NaN |
| 90000 | 177 | 7.6215E+141 | NaN | NaN |

Note that, while Table 4.8 shows the values of compound coefficient such as $A_k$, $B_k^1$, and $C_k^1$, Table 4.9 involves the particular parameters of the compound coefficient which is responsible for the breakdown. Therefore, it is potentially cheaper to check for breakdown in Table 4.9 than in Table 4.8. Similar tables for different instance can be found in Tables A.22-A.24, subsection A.2.4 of Appendix A. After gathering all these, thus, we finally obtain the following algorithm

**Algorithm 19** Monitoring Lanczos-type Algorithm based on relation $A_8/B_{10}$

**Input:** $A$ an $n \times n$ matrix, $b$ an n-vector.
**Output:** the approximations solution, $\mathbf{x}_k$, norm of the residual, $||\mathbf{r}_k||$.
Initializations: Choose $\mathbf{x}_0$ and $\mathbf{y}$, and the tolerance $\varepsilon$ to $1E-13$.
Set $\mathbf{r}_0 = b - A\mathbf{x}_0$; $\mathbf{y}_0 = \mathbf{y}$, $\mathbf{z}_0 = \mathbf{r}_0$;
**Compute:**
$\mathbf{y}_1 = A^T\mathbf{y}_0$; $A_1$ as in (4.16);
$k = 0$;
**While** $||\mathbf{r}_k|| > \varepsilon$ **do**
$\mathbf{y}_k = A^T\mathbf{y}_{k-1}$;
$A_k$, as in (4.16) and $C_k^1$, $B_k^1$ as in(4.21) respectively;
$\mathbf{r}_k$, $\mathbf{x}_k$ as in (4.15) and $z_k$ as in (4.20) respectively
/* Monitor Denominators: $A_k$, $B_k^1$, $C_k^1$, $a_k$, $b_k$. */
/* Design a test/rule. The test might be based on choosing a threshold value $\epsilon$, for instance, for that parameter in the coefficients which caused breakdown. */

**If** ($|a_k| \leq 1E-25$ or $|b_k| \leq 1E-25$);
display('Check Z1 ......');
**break**;
**End**;
**If** ($|b_k| \geq \alpha_i$)
display('Check Y1 ......');
**break**;
**End**;

where $\alpha_i = 1E+130$, $1.0E+150$, $1E+280$;
when $i = 1, 2, 3$, for $\delta = 0$, $0.2$, $5$ respectively;

**Algorithm 19** $A_8/B_{10}$ based algorithm(continued)

**If** $(|b_k| \leq 1.0E - 25)$;
  display('Check Z2 ......');
  **break**;
**End**;
**If** $(|b_k| \geq 1E + 270)$;
  display('Check Y2 ......');
  **break**;
**End**;

for $\delta = 8$;
$k = k + 1$;
**EndWhile**
Obtain the approximate solution as well as the residual norm;
$\text{sol}_{last} = \mathbf{x}_k$;
$\text{norm}_{last} = ||\mathbf{r}_k||$;
**Stop**.

### 4.4.5 Can a test be based on the number of iteration.?

By looking at the column of $k$ in Table 4.10, it is obvious $k$ changes little with the change in dimension of the matrix except in few cases. It is , therefore, possible to design a restarting test based on $k$. however, at least up to dimension 180000, about 30% of cases will be missed. A test that also includes the value of $A_k$ may remedy this shortcoming such a test may be as

**Test:**

maxval=300;

**If** (k $\geq$ maxval) or ($A_k == 0$) **Then**

  Restart;

**EndIf**

**Table 4.10:** *Behaviour of the parameters of the offending coefficients of $A_8/B_{10}$ on Baheux-type problems when $\delta = 0$*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $A_k$ | $B_k$ | $E_k$ |
| 100 | 110 | 0.0000E+00 | -Inf | 0.0000E+00 |
| 500 | 300 | 0.0000E+00 | Inf | 0.0000E+00 |
| 1000 | 354 | NaN | NaN | 2.7707E+01 |
| 5000 | 347 | 0.0000E+00 | -Inf | 0.0000E+00 |
| 10000 | 217 | 0.0000E+00 | Inf | 0.0000E+00 |
| 20000 | 354 | NaN | NaN | -1.2788E+02 |
| 30000 | 354 | NaN | NaN | 2.2656e-01 |
| 40000 | 354 | NaN | NaN | -5.5784E+01 |
| 50000 | 198 | 0.0000E+00 | Inf | 0.0000E+00 |
| 60000 | 354 | NaN | NaN | -2.1109E+00 |
| 70000 | 326 | 0.0000E+00 | Inf | 0.0000E+00 |
| 80000 | 354 | NaN | NaN | -1.0170E+01 |
| 90000 | 182 | 0.0000E+00 | -Inf | 0.0000E+00 |
| 100000 | 113 | 0.0000E+00 | Inf | 0.0000E+00 |
| 110000 | 354 | NaN | NaN | -1.3942E+01 |
| 120000 | 354 | NaN | NaN | -1.3913E+00 |
| 130000 | 354 | NaN | NaN | -5.5625E+00 |
| 140000 | 354 | NaN | NaN | -2.2225E+01 |
| 150000 | 354 | NaN | NaN | -5.3333E+00 |
| 160000 | 354 | NaN | NaN | -3.4702E+00 |
| 170000 | 354 | NaN | NaN | -1.3091E+01 |
| 180000 | 151 | 0.0000E+00 | -Inf | 0.0000E+00 |

## 4.5 Restarting Strategies

In these strategies, the idea is either to stop the Lancozs-type algorithm pre-emptively and restart it with some iterate or wait until breakdown occurs and then restart from the last iterate found. It is reasonable to restart from the point immediately before the breakdown occurred if one can detect it. Otherwise, one may consider restarting strategy after breakdown has happened [36]. Different strategies, **ST1**, **ST2** and **ST3**, can be used for restarting various algorithms as already explained in Section 1.8.1.

### 4.5.1 ST2 Implementation

ST2 takes as input a given algorithm from a prespecified list. Here, these algorithms are the ones already listed above, i.e. $A_4$, $A_{12}$, $A_8/B_6$, and $A_8/B_{10}$. Depending on whether the algorithms are of the $A_i$-type (i.e. Lanczos-type algorithm based on a single recurrence relation) or $A_i/B_j$-type (i.e. Lanczos-type algorithm based on two recurrence relations), initialisation has to be done differently; $A_i$-type requires $x_0$, $r_0 = b - Ax$ and $y_0 = y$, and $A_i/B_j$-type requires $x_0$, $r_0 = b - Ax$ and $y_0 = y$, as well as $z_0 = r_0$. The general ST2 algorithm can be described, therefore, as follows.

**Algorithm 20** Restarting Algorithm Based on Monitoring

Choose restarting strategy **ST2**.
{**Step 1**}
Start with Monitoring Lanczos-type algorithms from prespecified list
$\{Alg : 16, Alg : 17, Alg : 18, Alg : 19\}$.
{**Step 2**}
Run chosen Monitoring Lanczos-type algorithm until it halts;
Obtain the solution $sol_{last} = \mathbf{x}_k$ as well as the residual norm $norm_{last} = ||\mathbf{r}_k||$.
**While** $||\mathbf{r}_k|| > \varepsilon$ **do**
    Initialize it with the current iterate of the algorithm run;
    $\mathbf{x} = sol_{last}$,
    $\mathbf{y} = b - A\mathbf{x}$.
    Run chosen Monitoring algorithm;
**EndWhile**
Obtain the optimal solution as well as the optimal residual norm as follows
$sol_{optimal} = \mathbf{x}_k$
$norm_{optimal} = ||\mathbf{r}_k||$.
**Stop**.

## 4.6 Restarting Algorithm 17

The solution is obtained via restarting the Algorithm 17 as given in Algorithm 20. Utilizing regular intervals, the algorithm is restarted using the current iterate.

### 4.6.1 Numerical Results

The results obtained with Algorithm 12 and Algorithm 20, on Baheux-type problems of different dimensions, for different values of $\delta$ [3,4], are presented in Tables 4.11-4.14.

**Table 4.11:** *Results of Algorithm 12 and Algorithm 20 on Baheux-type problems when* $\delta = 0$

| | Algorithm 12 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 185 | 2 | 4.9751E-14 | 7.0910E-01 |
| 500 | NaN | | 906 | 6 | 9.7847E-14 | 1.1169E+00 |
| 1000 | NaN | | 964 | 6 | 9.1003E-14 | 1.5922E+00 |
| 5000 | NaN | | 1011 | 6 | 9.3487E-14 | 3.6608E+01 |
| 10000 | NaN | | 1085 | 7 | 9.9417E-14 | 7.4854E+01 |
| 20000 | NaN | | 988 | 6 | 9.9324E-14 | 6.9171E+02 |
| 30000 | NaN | | 1089 | 7 | 9.9248E-14 | 3.4193E+03 |
| 40000 | NaN | | 1082 | 7 | 7.5591E-14 | 2.5580E+03 |
| 50000 | NaN | | 1257 | 8 | 8.1885E-14 | 2.9318E+03 |
| 60000 | NaN | | 1303 | 8 | 8.4811E-14 | 7.2413E+03 |
| 70000 | NaN | | 1128 | 7 | 8.7667E-14 | 7.3412E+03 |
| 80000 | NaN | | 1120 | 7 | 9.9146E-14 | 6.5786E+03 |
| 90000 | NaN | | 1072 | 7 | 9.0707E-14 | 5.0874E+03 |

**Table 4.12:** *Results of Algorithm 12 and Algorithm 20 on Baheux-type problems when* $\delta = 0.2$

| | Algorithm 12 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 356 | 3 | 3.4466E-14 | 9.0114E-01 |
| 500 | NaN | | 862 | 6 | 8.6592E-14 | 1.1456E+00 |
| 1000 | NaN | | 1355 | 8 | 8.1958E-14 | 2.2509E+00 |
| 5000 | NaN | | 898 | 6 | 9.4303E-14 | 2.9377E+01 |
| 10000 | NaN | | 883 | 6 | 8.9325E-14 | 1.7968E+02 |
| 20000 | NaN | | 972 | 6 | 8.3356E-14 | 3.9526E+02 |
| 30000 | NaN | | 1038 | 7 | 8.1079E-14 | 1.3205E+03 |
| 40000 | NaN | | 1220 | 8 | 8.9458E-14 | 3.3931E+03 |
| 50000 | NaN | | 1069 | 7 | 7.5661E-14 | 3.6229E+03 |
| 60000 | NaN | | 1051 | 7 | 9.1927E-14 | 4.7638E+03 |
| 70000 | NaN | | 904 | 6 | 8.2881E-14 | 6.3346E+03 |
| 80000 | NaN | | 1065 | 7 | 7.2007E-14 | 5.3332E+03 |
| 90000 | NaN | | 949 | 7 | 8.7481E-14 | 4.4072E+03 |

**Table 4.13:** *Results of Algorithm 12 and Algorithm 20 on Baheux-type problems when δ = 5*

| | Algorithm 12 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 1610 | 6 | 8.7093E-14 | 1.5935E+00 |
| 500 | NaN | | 1582 | 6 | 8.0338E-14 | 6.2823E+00 |
| 1000 | NaN | | 1370 | 5 | 9.7527E-14 | 1.3913E+01 |
| 5000 | NaN | | 1302 | 5 | 7.1808E-14 | 9.1577E+01 |
| 10000 | NaN | | 1739 | 6 | 8.5618E-14 | 3.8901E+02 |
| 20000 | NaN | | 1153 | 4 | 9.3103E-14 | 1.4575E+03 |
| 30000 | NaN | | 1846 | 7 | 7.3421E-14 | 2.6401E+03 |
| 40000 | NaN | | 1442 | 5 | 8.2381E-14 | 2.2795E+03 |
| 50000 | NaN | | 1093 | 4 | 9.0785E-14 | 4.9594E+03 |
| 60000 | NaN | | 1438 | 5 | 9.1695E-14 | 6.8288E+03 |
| 70000 | NaN | | 2623 | 9 | 9.9138E-14 | 7.6795E+03 |
| 80000 | NaN | | 1110 | 4 | 9.8909E-14 | 5.2076E+03 |
| 90000 | NaN | | 1445 | 5 | 9.5266E-14 | 8.4268E+03 |

**Table 4.14:** *Results of Algorithm 12 and Algorithm 20 on Baheux-type problems when δ = 8*

| | Algorithm 12 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 1520 | 7 | 9.9093E-14 | 2.1651E+00 |
| 500 | NaN | | 2045 | 9 | 7.7985E-14 | 2.3571E+00 |
| 1000 | NaN | | 1489 | 6 | 9.1843E-14 | 3.2646E+01 |
| 5000 | NaN | | 2258 | 9 | 9.9646E-14 | 8.7535E+01 |
| 10000 | NaN | | 2379 | 10 | 8.1828E-14 | 3.3375E+02 |
| 20000 | NaN | | 1419 | 6 | 9.5256E-14 | 6.1979E+03 |
| 30000 | NaN | | 2735 | 11 | 9.9598E-14 | 2.4927E+03 |
| 40000 | NaN | | 2462 | 10 | 9.1988E-14 | 3.8832E+03 |
| 50000 | NaN | | 2658 | 11 | 9.6892E-14 | 3.5659E+03 |
| 60000 | NaN | | 5637 | 22 | 9.7572E-14 | 2.4112E+04 |
| 70000 | NaN | | 2024 | 8 | 8.1102E-14 | 7.5391E+03 |
| 80000 | NaN | | 4308 | 17 | 9.4350E-14 | 1.6636E+04 |
| 90000 | NaN | | 3812 | 15 | 7.5429E-14 | 1.7716E+04 |

## 4.7 Restarting Algorithm 16

The solution is obtained via restarting Algorithm 16 as given in Algorithm 20. Utilizing regular intervals, the algorithm is restarted using the current iterate.

### 4.7.1 Numerical Results

The results obtained with Algorithm 11, and Algorithm 20 described above, on Baheux-type problems of different dimensions, for different values of $\delta$ [3,4], are presented in Tables 4.15-4.18.

**Table 4.15:** *Results of Algorithm 11 and Algorithm 20 on Baheux-type problems when* $\delta = 0$

| | Algorithm 11 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 149 | 2 | 5.4429E-14 | 7.5120E-01 |
| 500 | NaN | | 916 | 8 | 9.8779E-14 | 1.8921E+00 |
| 1000 | NaN | | 783 | 7 | 7.8942E-14 | 1.8649E+00 |
| 5000 | NaN | | 1046 | 9 | 9.6411E-14 | 6.2534E+01 |
| 10000 | NaN | | 924 | 8 | 8.6591E-14 | 1.2378E+02 |
| 20000 | NaN | | 1036 | 9 | 9.0168E-14 | 1.2158E+03 |
| 30000 | NaN | | 1046 | 9 | 6.2128E-14 | 1.7086E+03 |
| 40000 | NaN | | 1368 | 11 | 8.5319E-14 | 2.9172E+03 |
| 50000 | NaN | | 1180 | 10 | 8.8686E-14 | 5.6647E+03 |
| 60000 | NaN | | 1056 | 9 | 9.6952E-14 | 7.0835E+03 |
| 70000 | NaN | | 1013 | 8 | 9.9118E-14 | 9.3068E+03 |
| 80000 | NaN | | 919 | 8 | 9.7447E-14 | 6.9428E+03 |
| 90000 | NaN | | 936 | 8 | 9.3677E-14 | 8.8362E+03 |

**Table 4.16:** *Results of Algorithm 11 and Algorithm 20 on Baheux-type problems when* $\delta = 0.2$

| | Algorithm 11 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 352 | 4 | 8.4201E-14 | 9.3534E-01 |
| 500 | NaN | | 743 | 6 | 8.2836E-14 | 1.6902E+00 |
| 1000 | NaN | | 737 | 6 | 9.6689E-14 | 2.8613E+00 |
| 5000 | NaN | | 1159 | 9 | 8.7238E-14 | 4.1919E+01 |
| 10000 | NaN | | 884 | 7 | 9.3045E-14 | 1.2608E+02 |
| 20000 | NaN | | 853 | 7 | 9.3119E-14 | 7.3278E+02 |
| 30000 | NaN | | 1053 | 8 | 8.6376E-14 | 2.0646E+03 |
| 40000 | NaN | | 812 | 7 | 7.7838E-14 | 2.9067E+03 |
| 50000 | NaN | | 867 | 7 | 7.8088E-14 | 3.8596E+03 |
| 60000 | NaN | | 995 | 8 | 9.7165E-14 | 8.0922E+03 |
| 70000 | NaN | | 868 | 7 | 9.1179E-14 | 8.8500E+03 |
| 80000 | NaN | | 966 | 7 | 9.4984E-14 | 9.5485E+03 |
| 90000 | NaN | | 905 | 7 | 8.4068E-14 | 1.1762E+04 |

**Table 4.17:** *Results of Algorithm 11 and Algorithm 20 on Baheux-type problems when δ = 5*

| | Algorithm 11 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 802 | 8 | 7.3557E-14 | 1.0901E+00 |
| 500 | NaN | | 717 | 7 | 7.4404E-14 | 1.5427E+00 |
| 1000 | NaN | | 895 | 9 | 7.7603E-14 | 1.9029E+00 |
| 5000 | NaN | | 1027 | 10 | 8.8162E-14 | 7.0547E+01 |
| 10000 | NaN | | 2576 | 23 | 6.6752E-14 | 9.0615E+02 |
| 20000 | NaN | | 1842 | 17 | 5.0440E-14 | 1.9670E+03 |
| 30000 | NaN | | 1568 | 14 | 7.8304E-14 | 2.1650E+03 |
| 40000 | NaN | | 6878 | 59 | 8.3005E-14 | 1.6076E+04 |
| 50000 | NaN | | 1505 | 14 | 8.3499E-14 | 6.1923E+03 |
| 60000 | NaN | | 2180 | 20 | 9.6363E-14 | 1.2218E+04 |
| 70000 | NaN | | 3007 | 27 | 6.6189E-14 | 2.8736E+04 |
| 80000 | NaN | | 1059 | 10 | 7.0086E-14 | 1.1031E+04 |
| 90000 | NaN | | 1990 | 18 | 9.4227E-14 | 2.0319E+04 |

**Table 4.18:** *Results of Algorithm 11 and Algorithm 20 on Baheux-type problems when δ = 8*

| | Algorithm 11 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 1036 | 11 | 9.7360E-14 | 1.3808E+00 |
| 500 | NaN | | 1139 | 12 | 9.8301E-14 | 1.5955E+00 |
| 1000 | NaN | | 1273 | 13 | 9.4397E-14 | 3.7262E+00 |
| 5000 | NaN | | 1607 | 16 | 8.6392E-14 | 1.0017E+02 |
| 10000 | NaN | | 1991 | 20 | 7.6799E-14 | 5.9519E+02 |
| 20000 | NaN | | 3216 | 31 | 8.7604E-14 | 3.0332E+03 |
| 30000 | NaN | | 7017 | 66 | 4.2110E-14 | 1.2230E+04 |
| 40000 | NaN | | 6411 | 61 | 6.3927E-14 | 1.9988E+04 |
| 50000 | NaN | | 9148 | 86 | 6.9372E-14 | 2.5205E+04 |
| 60000 | NaN | | 1350 | 14 | 7.6603E-14 | 9.4175E+03 |
| 70000 | NaN | | 12687 | 119 | 7.8368E-14 | 1.0022E+05 |
| 80000 | NaN | | 1463 | 15 | 9.5044E-14 | 1.4259E+04 |
| 90000 | NaN | | 1684 | 17 | 9.9384E-14 | 1.7534E+04 |

## 4.8 Restarting Algorithm 18

The solution is obtained via restarting Algorithm 18 as given in Algorithm 20. Utilizing regular intervals, the algorithm is restarted using the current iterate.

### 4.8.1 Numerical Results

The results obtained with Algorithm 14 and its restarting version Algorithm 20, on Baheux-type problems of different dimensions, for different values of $\delta$ [3,4], are presented in Tables 4.19-4.22.

**Table 4.19:** *Results of Algorithm 14 and Algorithm 20 on Baheux-type problems when* $\delta = 0$

| | Algorithm 14 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 57 | 2 | 6.1115E-14 | 6.2013E-01 |
| 500 | NaN | | 529 | 9 | 9.8912E-14 | 5.3377E+00 |
| 1000 | NaN | | 1060 | 15 | 8.5325E-14 | 2.8432E+00 |
| 5000 | NaN | | 611 | 9 | 9.0619E-14 | 1.5942E+01 |
| 10000 | NaN | | 612 | 9 | 8.7698E-14 | 6.3695E+01 |
| 20000 | NaN | | 916 | 13 | 9.6487E-14 | 3.2046E+02 |
| 30000 | NaN | | 763 | 11 | 9.7053E-14 | 4.6949E+02 |
| 40000 | NaN | | 922 | 13 | 9.7491E-14 | 1.2360E+03 |
| 50000 | NaN | | 766 | 11 | 8.7656E-14 | 1.2205E+03 |
| 60000 | NaN | | 679 | 10 | 8.7424E-14 | 1.5603E+03 |
| 70000 | NaN | | 633 | 9 | 9.0205E-14 | 1.8936E+03 |
| 80000 | NaN | | 706 | 10 | 9.8981E-14 | 2.7773E+03 |
| 90000 | NaN | | 830 | 12 | 8.4513E-14 | 4.3607E+03 |

**Table 4.20:** *Results of Algorithm 14 and Algorithm 20 on Baheux-type problems when* $\delta = 0.2$

| | Algorithm 14 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 689 | 8 | 7.7436E-14 | 8.4320E-01 |
| 500 | NaN | | 1463 | 16 | 5.9065E-14 | 1.5156E+00 |
| 1000 | NaN | | 1359 | 15 | 7.9484E-14 | 2.4188E+00 |
| 5000 | NaN | | 1717 | 19 | 9.7027E-14 | 7.6450E+01 |
| 10000 | NaN | | 1256 | 14 | 8.6415E-14 | 2.1527E+02 |
| 20000 | NaN | | 897 | 10 | 5.8727E-14 | 4.2976E+02 |
| 30000 | NaN | | 1634 | 18 | 9.6493E-14 | 1.7039E+03 |
| 40000 | NaN | | 1155 | 13 | 8.3941E-14 | 2.5913E+03 |
| 50000 | NaN | | 1564 | 17 | 7.4461E-14 | 3.9558E+03 |
| 60000 | NaN | | 1249 | 14 | 8.6231E-14 | 4.7694E+03 |
| 70000 | NaN | | 1207 | 14 | 7.7883E-14 | 5.7245E+03 |
| 80000 | NaN | | 1896 | 21 | 7.9999E-14 | 8.5143E+03 |
| 90000 | NaN | | 2231 | 24 | 9.2184E-14 | 1.4880E+04 |

**Table 4.21:** *Results of Algorithm 14 and Algorithm 20 on Baheux-type problems when δ = 5*

| | Algorithm 14 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 686 | 5 | 9.0831E-14 | 8.3733E-01 |
| 500 | NaN | | 777 | 6 | 9.0251E-14 | 1.7225E+00 |
| 1000 | NaN | | 1065 | 8 | 5.6323E-14 | 3.8374E+00 |
| 5000 | NaN | | 1081 | 8 | 6.4990E-14 | 4.8549E+01 |
| 10000 | NaN | | 1360 | 10 | 8.6696E-14 | 2.1502E+02 |
| 20000 | NaN | | 1509 | 11 | 5.7739E-14 | 7.5752E+02 |
| 30000 | NaN | | 1518 | 11 | 7.5207E-14 | 1.6380E+03 |
| 40000 | NaN | | 1370 | 10 | 7.2925E-14 | 2.8792E+03 |
| 50000 | NaN | | 1246 | 9 | 6.8686E-14 | 3.5663E+03 |
| 60000 | NaN | | 1068 | 8 | 5.5732E-14 | 4.5924E+03 |
| 70000 | NaN | | 1374 | 10 | 8.4638E-14 | 6.2884E+03 |
| 80000 | NaN | | 1221 | 9 | 7.0494E-14 | 6.2962E+03 |
| 90000 | NaN | | 1373 | 10 | 9.0335E-14 | 6.3175E+03 |

**Table 4.22:** *Results of Algorithm 14 and Algorithm 20 on Baheux-type problems when δ = 8*

| | Algorithm 14 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 827 | 7 | 8.7952E-14 | 9.5710E-01 |
| 500 | NaN | | 961 | 8 | 5.5997E-14 | 1.9370E+00 |
| 1000 | NaN | | 1083 | 9 | 9.5325E-14 | 1.9542E+00 |
| 5000 | NaN | | 1217 | 10 | 9.5816E-14 | 3.4618E+01 |
| 10000 | NaN | | 1227 | 10 | 9.8573E-14 | 1.1067E+02 |
| 20000 | NaN | | 1487 | 12 | 7.3681E-14 | 5.3876E+02 |
| 30000 | NaN | | 1363 | 11 | 4.6276E-14 | 1.0301E+03 |
| 40000 | NaN | | 1247 | 10 | 9.0025E-14 | 1.5393E+03 |
| 50000 | NaN | | 1616 | 13 | 9.1182E-14 | 3.2238E+03 |
| 60000 | NaN | | 1224 | 10 | 4.9347E-14 | 3.7250E+03 |
| 70000 | NaN | | 1359 | 11 | 7.1960E-14 | 4.0946E+03 |
| 80000 | NaN | | 1617 | 13 | 9.0161E-14 | 5.9952E+03 |
| 90000 | NaN | | 1362 | 11 | 5.8420E-14 | 6.4207E+03 |

## 4.9 Restarting Algorithm 19

The solution is obtained via restarting Algorithm 19 as given in Algorithm 20. Utilizing regular intervals, the algorithm is restarted using the current iterate.

### 4.9.1 Numerical results

The results obtained with Algorithm 13 and its restarting version Algorithm 20, on Baheux-type problems of different dimensions, for different values of $\delta$ [3,4], are presented in Tables 4.23-4.26.

**Table 4.23:** *Results of Algorithm 13 and Algorithm 20 on Baheux-type problems when* $\delta = 0$

| | Algorithm 13 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 665 | 7 | 9.3606E-14 | 8.9790E-01 |
| 500 | NaN | | 3776 | 40 | 8.8988E-14 | 5.1838E+00 |
| 1000 | NaN | | 5343 | 57 | 9.3373E-14 | 2.5808E+01 |
| 5000 | NaN | | 4173 | 44 | 9.3051E-14 | 4.6781E+02 |
| 10000 | NaN | | 3027 | 33 | 9.1131E-14 | 1.1683E+03 |
| 20000 | NaN | | 1256 | 14 | 8.7733E-14 | 1.9221E+03 |
| 30000 | NaN | | 1186 | 13 | 7.9795E-14 | 4.0962E+03 |
| 40000 | NaN | | 2269 | 26 | 9.5148E-14 | 1.2473E+04 |
| 50000 | NaN | | 2107 | 23 | 9.9344E-14 | 1.8289E+03 |
| 60000 | NaN | | 2751 | 29 | 9.5721E-14 | 2.8230E+04 |
| 70000 | NaN | | 4505 | 49 | 9.1219E-14 | 6.4745E+04 |
| 80000 | NaN | | 1925 | 21 | 8.9973E-14 | 3.7393E+04 |
| 90000 | NaN | | 2448 | 27 | 9.0799E-14 | 5.8112E+04 |

**Table 4.24:** *Results of Algorithm 13 and Algorithm 20 on Baheux-type problems when* $\delta = 0.2$

| | Algorithm 13 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 885 | 9 | 9.3360E-14 | 9.8385E-01 |
| 500 | NaN | | 4965 | 47 | 9.6024E-14 | 1.3427E+01 |
| 1000 | NaN | | 2517 | 25 | 8.2166E-14 | 1.9866E+01 |
| 5000 | NaN | | 991 | 10 | 8.7901E-14 | 1.2460E+02 |
| 10000 | NaN | | 1865 | 19 | 9.9181E-14 | 8.3228E+02 |
| 20000 | NaN | | 1463 | 15 | 9.5309E-14 | 2.4773E+03 |
| 30000 | NaN | | 2593 | 26 | 7.9825E-14 | 1.0237E+04 |
| 40000 | NaN | | 2211 | 21 | 8.9642E-14 | 1.4145E+04 |
| 50000 | NaN | | 1521 | 15 | 8.1724E-14 | 1.1617E+04 |
| 60000 | NaN | | 1299 | 13 | 9.9193E-14 | 1.6511E+04 |
| 70000 | NaN | | 2784 | 27 | 8.4839E-14 | 4.0159E+04 |
| 80000 | NaN | | 2181 | 21 | 8.9733E-14 | 4.2025E+04 |
| 90000 | NaN | | 912 | 9 | 8.7492E-14 | 2.3436E+04 |

**Table 4.25:** *Results of Algorithm 13 and Algorithm 20 on Baheux-type problems when δ = 5*

| | Algorithm 13 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 633 | 5 | 7.5452E-14 | 8.0537E-01 |
| 500 | NaN | | 953 | 7 | 8.8992E-14 | 1.9199E+00 |
| 1000 | NaN | | 1218 | 9 | 8.5448E-14 | 9.9276E+00 |
| 5000 | NaN | | 1379 | 10 | 6.4556E-14 | 1.6484E+01 |
| 10000 | NaN | | 1377 | 10 | 6.5695E-14 | 6.7022E+02 |
| 20000 | NaN | | 1510 | 11 | 8.0851E-14 | 2.7845E+03 |
| 30000 | NaN | | 1507 | 11 | 9.0990E-14 | 6.0153E+03 |
| 40000 | NaN | | 1806 | 13 | 7.9073E-14 | 1.2243E+04 |
| 50000 | NaN | | 2093 | 15 | 9.2173E-14 | 1.9044E+04 |
| 60000 | NaN | | 2572 | 18 | 9.5965E-14 | 2.9860E+04 |
| 70000 | NaN | | 1814 | 13 | 4.4036E-14 | 3.2972E+04 |
| 80000 | NaN | | 1800 | 13 | 5.1310E-14 | 4.9652E+04 |
| 90000 | NaN | | 1840 | 13 | 3.2006E-14 | 4.3166E+04 |

**Table 4.26:** *Results of Algorithm 13 and Algorithm 20 on Baheux-type problems when δ = 8*

| | Algorithm 13 | | Algorithm 20 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit [1] | Cycles[2] | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\|r_k\|\|$ | t(sec) |
| 100 | NaN | | 817 | 7 | 9.2905E-14 | 8.4382E-01 |
| 500 | NaN | | 1035 | 9 | 8.1199E-14 | 2.7530E+00 |
| 1000 | NaN | | 1144 | 10 | 8.3085E-14 | 5.4307E+00 |
| 5000 | NaN | | 1758 | 15 | 9.1337E-14 | 1.9199E+02 |
| 10000 | NaN | | 1408 | 12 | 6.6426E-14 | 5.4412E+02 |
| 20000 | NaN | | 1267 | 11 | 8.4975E-14 | 1.5009E+03 |
| 30000 | NaN | | 1777 | 15 | 8.6743E-14 | 4.9905E+03 |
| 40000 | NaN | | 1773 | 15 | 9.5829E-14 | 8.518E+03 |
| 50000 | NaN | | 1654 | 14 | 6.6637E-14 | 1.5988E+04 |
| 60000 | NaN | | 1643 | 14 | 5.2675E-14 | 2.3131E+04 |
| 70000 | NaN | | 1533 | 13 | 7.9407E-14 | 2.5631E+04 |
| 80000 | NaN | | 1395 | 12 | 7.0229E-14 | 2.7149E+04 |
| 90000 | NaN | | 1711 | 14 | 8.0416E-14 | 4.1674E+04 |

[1] add all the number of iteration during each cycle
[2] A cycle is a number of iterations carried out in a restart or a switch.

## 4.10 Comments

These tests and tables prove that it is possible to have a more targeted number of cycles (switches/ restarts) and their lengths. To illustrate, consider a similar test in Maharani [55] shows that our approach leads to a more efficient and robust Lanczos-type algorithm implementation.

## 4.11 Summary

The restarting strategies **ST2** and **ST3** used in this work are successful in handling the breakdown in Lanczos-type algorithms. This is supported by strong numerical evidence. They successfully solved problems with dimensions up to 90000 whereas individual algorithms with no restarting facility could only solve problems with dimensions $\leq 30$. Moreover, the cost involved in such preemptive restarting is not very high. Monitoring the coefficients that can approach zero, has a cost which is similar to that of a test of the form "if |MonitorDenom value| ≤ tolerance-then stop". Many such tests could be done by using various tolerance levels. It impact on the overall computing time has not been measured in this thesis. Favourable results hint to restarting as a useful approach to handling breakdown while solving SLE's by Lanczos-type algorithms. The idea not only differs from existing strategies for handling breakdowns, [12, 15, 18], but it is also simple to understand and use. Further extensive testing needs to be done on both large real and randomly generated problems to get a complete picture of the behavior and cost of the restarting approach in comparison to state-of-the-art Lanczos-type algorithms.

# Chapter 5

# Switching between Lanczos-type algorithms to avoid breakdown

This chapter is devoted to the switching strategy to avoid the issue of breakdown in Lanczos-type algorithms that arises due to the non-existence of some coefficients of the recurrence relations that provide a base for the algorithms. The non-existence of coefficients for Lanczos-type algorithm on a specific iterate of the recurrence relations does not, necessarily, cause the problem for another Lanczos-type algorithm, based on different recurrence relations. It, thus, follows that one might switch to other algorithms to avoid breakdown. This allows to carry on in a Krylov space having a different basis. It, therefore, could be concluded that switching might be considered as a potential remedy for the breakdown issues [33].

## 5.1 Switching Algorithm

A set of Lanczos-type algorithms can be switched from one algorithm to another using strategies **ST1**, **ST2** or **ST3** as given in Section 1.8.1. Note that in the last cycle, if the chosen algorithm is the same as the one running in the first cycle, then it is a case of restarting.

Otherwise, it is switching.

**Algorithm 21** Switching Algorithm Based on Monitoring

{**Step 1**}
Choose a strategy ST2.
Start with Monitoring Lanczos-type algorithms from prespecified list
$\{Alg:16, Alg:17, Alg:18, Alg:19\}$.
{**Step 2**}
Run algorithm until it halts;
**If** solution is obtained **Then**
  **Stop**;
**Else**
  Switch to another algorithm;
  Initialize it with current iterate of the algorithm running in the last cycle;
  $\mathbf{x} = sol_{last}$;
  $\mathbf{y} = b - A\mathbf{x}$;
  go to Step 2;
**EndIf**
Obtain the optimal solution as well as the optimal residual norm as follows
$sol_{optimal} = \mathbf{x}_k$
$norm_{optimal} = ||\mathbf{r}_k||$.
**Stop**.

## 5.2 Switching between Algorithm 16 and Algorithm 17

Algorithm 21, starts with either Algorithm 16 or Algorithm 17. Then it is halted before breakdown, and switching to the other is carried out.

### 5.2.1 Numerical Results

The switching procedure between Algorithms 16 and Algorithm 17 has been implemented in Matlab and applied to Baheux-type problems of different dimensions, for different values of $\delta = 0, 0.2, 5$ and 8. These problems have been described in [3, 4]. The dimension of the coefficient matrix $A$ is $n = n_1 \times n_2$, where $n_1$ is the number of block matrices in $A$ and $n_2$ is the dimension of the matrix $B$ which is fixed to 10. The results obtained with Algorithm 11, Algorithm 12 and the switching Algorithm 21, are presented in Tables 5.1-5.4.

**Table 5.1:** *Results of Algorithm 11, Algorithm 12 and Algorithm 21 on Baheux-type problems when* $\delta = 0$

| | Algorithm 11 | Algorithm 12 | Algorithm 21 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | $\|r_k\|$ | $\sum k$ | | $\| r_k \|$ | sec |
| 100 | NaN | NaN | 142 | 2 | 6.4799E-14 | 1.0229E+00 |
| 500 | NaN | NaN | 858 | 6 | 9.9621E-14 | 1.3673E+00 |
| 1000 | NaN | NaN | 926 | 7 | 8.3930E-14 | 1.5031E+00 |
| 5000 | NaN | NaN | 1011 | 6 | 9.3487E-14 | 3.0330E+01 |
| 10000 | NaN | NaN | 1064 | 8 | 9.2032E-14 | 1.3467E+02 |
| 20000 | NaN | NaN | 1009 | 7 | 9.1714E-14 | 3.7645E+02 |
| 30000 | NaN | NaN | 1131 | 8 | 9.5415E-14 | 8.7870E+02 |
| 40000 | NaN | NaN | 1181 | 8 | 9.8820E-14 | 1.5565E+03 |
| 50000 | NaN | NaN | 1312 | 9 | 9.9651E-14 | 2.6257E+03 |
| 60000 | NaN | NaN | 972 | 7 | 9.4797E-14 | 2.3296E+03 |
| 70000 | NaN | NaN | 1146 | 8 | 8.0121E-14 | 3.9998E+03 |
| 80000 | NaN | NaN | 1071 | 8 | 8.0435E-14 | 5.7842E+03 |
| 90000 | NaN | NaN | 1072 | 7 | 9.0707E-14 | 4.6034E+03 |

**Table 5.2:** *Results of Algorithm 11, Algorithm 12 and Algorithm 21 on Baheux-type problems when* $\delta = 0.2$

| | Algorithm 11 | Algorithm 12 | Algorithm 21 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | $\|r_k\|$ | $\sum k$ | | $\| r_k \|$ | sec |
| 100 | NaN | NaN | 409 | 3 | 7.2018E-14 | 1.7407E+00 |
| 500 | NaN | NaN | 1015 | 7 | 6.3987E-14 | 1.1266E+00 |
| 1000 | NaN | NaN | 1087 | 8 | 8.2681E-14 | 2.0396E+00 |
| 5000 | NaN | NaN | 886 | 7 | 9.3962E-14 | 2.4518E+01 |
| 10000 | NaN | NaN | 913 | 7 | 9.3596E-14 | 8.1512E+01 |
| 20000 | NaN | NaN | 1236 | 8 | 8.6842E-14 | 7.7655E+02 |
| 30000 | NaN | NaN | 1160 | 8 | 9.2112E-14 | 1.1766E+03 |
| 40000 | NaN | NaN | 1588 | 10 | 8.4260E-14 | 2.7420E+03 |
| 50000 | NaN | NaN | 938 | 7 | 9.5880E-14 | 2.6978E+03 |
| 60000 | NaN | NaN | 1017 | 7 | 9.6912E-14 | 2.8531E+03 |
| 70000 | NaN | NaN | 949 | 7 | 9.6283E-14 | 3.5907E+03 |
| 80000 | NaN | NaN | 1041 | 7 | 9.5650E-14 | 6.0355E+03 |
| 90000 | NaN | NaN | 818 | 6 | 8.3997E-14 | 5.0977E+03 |

**Table 5.3:** *Results of Algorithm 11, Algorithm 12 and Algorithm 21 on Baheux-type problems when* $\delta = 5$

| | Algorithm 11 | Algorithm 12 | Algorithm 21 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | $\|r_k\|$ | $\sum k$ | | $\| r_k \|$ | sec |
| 100 | NaN | NaN | 661 | 5 | 6.6466E-14 | 1.4785E+00 |
| 500 | NaN | NaN | 1161 | 8 | 7.5564E-14 | 3.3436E+00 |
| 1000 | NaN | NaN | 1182 | 6 | 5.5675E-14 | 2.8682E+00 |
| 5000 | NaN | NaN | 1141 | 6 | 8.8864E-14 | 4.1462E+01 |
| 10000 | NaN | NaN | 972 | 5 | 8.6003E-14 | 1.2266E+02 |
| 20000 | NaN | NaN | 1577 | 9 | 8.7003E-14 | 8.5757E+02 |
| 30000 | NaN | NaN | 1339 | 6 | 9.5774E-14 | 1.0993E+03 |
| 40000 | NaN | NaN | 1971 | 10 | 4.6795E-14 | 2.6667E+03 |
| 50000 | NaN | NaN | 923 | 4 | 7.0378E-14 | 3.1699E+03 |
| 60000 | NaN | NaN | 1693 | 8 | 9.6513E-14 | 7.2279E+03 |
| 70000 | NaN | NaN | 2074 | 8 | 9.1983E-14 | 9.2871E+03 |
| 80000 | NaN | NaN | 3250 | 17 | 9.6659E-14 | 1.6062E+04 |
| 90000 | NaN | NaN | 5744 | 26 | 8.3977E-14 | 3.1191E+04 |

**Table 5.4:** *Results of Algorithm 11, Algorithm 12 and Algorithm 21 on Baheux-type problems when* $\delta = 8$

| | Algorithm 11 | Algorithm 12 | Algorithm 21 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | $\|r_k\|$ | $\sum k$ | | $\| r_k \|$ | sec |
| 100 | NaN | NaN | 1128 | 7 | 8.1255E-14 | 2.3917E+00 |
| 500 | NaN | NaN | 1178 | 8 | 9.8984E-14 | 3.0222E+00 |
| 1000 | NaN | NaN | 1359 | 8 | 8.8030E-14 | 7.9943E+00 |
| 5000 | NaN | NaN | 1298 | 9 | 7.7964E-14 | 8.0938E+01 |
| 10000 | NaN | NaN | 1855 | 10 | 9.9435E-14 | 3.8959E+02 |
| 20000 | NaN | NaN | 1700 | 9 | 7.2781E-14 | 1.0799E+03 |
| 30000 | NaN | NaN | 1442 | 9 | 6.8522E-14 | 2.0707E+03 |
| 40000 | NaN | NaN | 2248 | 12 | 9.9335E-14 | 3.9593E+03 |
| 50000 | NaN | NaN | 2254 | 15 | 9.6901E-14 | 8.6370E+03 |
| 60000 | NaN | NaN | 2405 | 15 | 6.6847E-14 | 8.7347E+03 |
| 70000 | NaN | NaN | 1752 | 13 | 8.6180E-14 | 7.9854E+03 |
| 80000 | NaN | NaN | 1361 | 8 | 9.3185E-14 | 6.6251E+03 |
| 90000 | NaN | NaN | 1894 | 10 | 9.7444E-14 | 1.0476E+04 |

## 5.3 Switching between Algorithm 17 and Algorithm 18

Algorithm 21 is started with either Algorithm 17 or Algorithm 18. The started algorithm after breakdown is switched to either of the two Algorithm 17 or Algorithm 18 chosen randomly.

### 5.3.1 Numerical Results

The results obtained with Algorithm 12, Algorithm 14 and the switching Algorithm 21 on Baheux-type problems of different dimensions, for different values of $\delta$ are shown in Tables 5.5-5.8.

**Table 5.5:** *Results of Algorithm 12, Algorithm 14 and Algorithm 21 on Baheux-type problems when* $\delta = 0$

| | Algorithm 12 | Algorithm 14 | Algorithm 21 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | $\|r_k\|$ | $\sum k$ | | $\| r_k \|$ | sec |
| 100 | NaN | NaN | 185 | 2 | 4.9961E-14 | 3.4382E+00 |
| 500 | NaN | NaN | 911 | 6 | 9.0453E-14 | 2.9074E+00 |
| 1000 | NaN | NaN | 874 | 7 | 7.2280E-14 | 2.8669E+00 |
| 5000 | NaN | NaN | 858 | 8 | 9.6192E-14 | 2.1875E+01 |
| 10000 | NaN | NaN | 1119 | 9 | 9.5648E-14 | 1.2321E+02 |
| 20000 | NaN | NaN | 1420 | 10 | 8.1789E-14 | 5.3626E+02 |
| 30000 | NaN | NaN | 997 | 9 | 9.0673E-14 | 7.8963E+02 |
| 40000 | NaN | NaN | 1038 | 9 | 8.8592E-14 | 1.4397E+03 |
| 50000 | NaN | NaN | 841 | 8 | 9.6474E-14 | 2.0097E+03 |
| 60000 | NaN | NaN | 911 | 9 | 8.3200E-14 | 2.3240E+03 |
| 70000 | NaN | NaN | 981 | 9 | 8.7603E-14 | 3.6608E+03 |
| 80000 | NaN | NaN | 983 | 7 | 7.4715E-14 | 3.7517E+03 |
| 90000 | NaN | NaN | 890 | 7 | 8.8803E-14 | 4.0620E+03 |

**Table 5.6:** *Results of Algorithm 12, Algorithm 14 and Algorithm 21 on Baheux-type problems when* $\delta = 0.2$

| | Algorithm 12 | Algorithm 14 | Algorithm 21 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | $\|r_k\|$ | $\sum k$ | | $\| r_k \|$ | sec |
| 100 | NaN | NaN | 356 | 3 | 3.3645E-14 | 1.2079E+00 |
| 500 | NaN | NaN | 783 | 6 | 7.1319E-14 | 1.7391E+00 |
| 1000 | NaN | NaN | 896 | 7 | 8.7500E-14 | 2.8320E+00 |
| 5000 | NaN | NaN | 995 | 9 | 5.8100E-14 | 4.5240E+01 |
| 10000 | NaN | NaN | 693 | 7 | 8.3533E-14 | 7.8762E+01 |
| 20000 | NaN | NaN | 906 | 8 | 9.9500E-14 | 3.8528E+02 |
| 30000 | NaN | NaN | 975 | 8 | 7.7623E-14 | 1.3711E+03 |
| 40000 | NaN | NaN | 1152 | 8 | 9.8596E-14 | 1.2034E+03 |
| 50000 | NaN | NaN | 1181 | 10 | 9.3551E-14 | 1.8797E+03 |
| 60000 | NaN | NaN | 1351 | 11 | 8.6233E-14 | 2.9447E+03 |
| 70000 | NaN | NaN | 1071 | 9 | 7.2445E-14 | 5.1275E+03 |
| 80000 | NaN | NaN | 891 | 7 | 8.8917E-14 | 4.8429E+03 |
| 90000 | NaN | NaN | 1074 | 8 | 9.2474E-14 | 4.7860E+03 |

**Table 5.7:** *Results of Algorithm 12, Algorithm 14 and Algorithm 21 on Baheux-type problems when δ = 5*

| | Algorithm 12 | Algorithm 14 | Algorithm 21 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | $\|r_k\|$ | $\sum k$ | | $\| r_k \|$ | sec |
| 100 | NaN | NaN | 658 | 5 | 8.7404E-14 | 1.2224E+00 |
| 500 | NaN | NaN | 1109 | 5 | 9.1274E-14 | 1.5485E+00 |
| 1000 | NaN | NaN | 1194 | 6 | 9.5098E-14 | 2.5360E+00 |
| 5000 | NaN | NaN | 1264 | 5 | 9.9088E-14 | 3.9120E+01 |
| 10000 | NaN | NaN | 1368 | 7 | 4.2928E-14 | 1.2078E+02 |
| 20000 | NaN | NaN | 1153 | 4 | 9.3103E-14 | 5.4273E+02 |
| 30000 | NaN | NaN | 1479 | 7 | 8.9283E-14 | 1.3146E+03 |
| 40000 | NaN | NaN | 1906 | 7 | 9.3332E-14 | 3.9116E+03 |
| 50000 | NaN | NaN | 1072 | 6 | 7.3005E-14 | 2.7799E+03 |
| 60000 | NaN | NaN | 1056 | 5 | 9.8609E-14 | 2.7222E+03 |
| 70000 | NaN | NaN | 1469 | 7 | 5.9755E-14 | 5.9907E+03 |
| 80000 | NaN | NaN | 1050 | 5 | 9.8520E-14 | 5.0974E+03 |
| 90000 | NaN | NaN | 1376 | 6 | 9.2455E-14 | 6.1762E+03 |

**Table 5.8:** *Results of Algorithm 12, Algorithm 14 and Algorithm 21 on Baheux-type problems when δ = 8*

| | Algorithm 12 | Algorithm 14 | Algorithm 21 | | | |
|---|---|---|---|---|---|---|
| Prob. size | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | $\|r_k\|$ | $\sum k$ | | $\| r_k \|$ | sec |
| 100 | NaN | NaN | 1011 | 6 | 9.0275E-14 | 1.4645E+00 |
| 500 | NaN | NaN | 1374 | 7 | 9.2989E-14 | 1.5570E+00 |
| 1000 | NaN | NaN | 1179 | 7 | 8.5810E-14 | 1.5081E+00 |
| 5000 | NaN | NaN | 1223 | 8 | 9.5828E-14 | 4.9891E+01 |
| 10000 | NaN | NaN | 1425 | 8 | 9.8665E-14 | 9.7959E+01 |
| 20000 | NaN | NaN | 1723 | 8 | 8.2101E-14 | 6.9990E+02 |
| 30000 | NaN | NaN | 1542 | 8 | 7.5286E-14 | 1.4793E+03 |
| 40000 | NaN | NaN | 1591 | 9 | 8.4668E-14 | 3.8185E+03 |
| 50000 | NaN | NaN | 1684 | 11 | 8.2775E-14 | 4.3103E+03 |
| 60000 | NaN | NaN | 1566 | 8 | 5.3064E-14 | 4.0704E+03 |
| 70000 | NaN | NaN | 1810 | 9 | 7.6455E-14 | 4.5509E+03 |
| 80000 | NaN | NaN | 1898 | 10 | 7.9890E-14 | 8.6573E+03 |
| 90000 | NaN | NaN | 1490 | 9 | 7.7909E-14 | 7.1199E+03 |

## 5.4 Switching between Algorithm 17 and Algorithm 19

Algorithm 21 is started with either Algorithm 17 or Algorithm 19, i.e. one of the algorithms run and halted before breakdown and then the switch to either of them chosen randomly is carried out.

### 5.4.1 Numerical Results

The results obtained with Algorithm 12, Algorithm 13 and the switching Algorithm 21 on Baheux-type problems of different dimensions, for different values of $\delta$ are shown in Tables 5.9-5.12.

**Table 5.9:** *Results of Algorithm 12, Algorithm 13 and Algorithm 21 on Baheux-type problems when* $\delta = 0$

| | Algorithm 12 | | Algorithm 13 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\| r_k \|\|$ | t(sec) |
| 100 | NaN | | NaN | | 185 | 2 | 4.9751E-14 | 7.4636E-01 |
| 500 | NaN | | NaN | | 822 | 6 | 6.2396E-14 | 3.7712E+00 |
| 1000 | NaN | | NaN | | 1223 | 9 | 9.2338E-14 | 2.3264E+01 |
| 5000 | NaN | | NaN | | 1026 | 9 | 9.8010E-14 | 5.1871E+01 |
| 10000 | NaN | | NaN | | 1089 | 8 | 9.5115E-14 | 2.0564E+02 |
| 20000 | NaN | | NaN | | 938 | 8 | 9.9848E-14 | 1.5493E+03 |
| 30000 | NaN | | NaN | | 1448 | 11 | 9.5437E-14 | 3.5232E+03 |
| 40000 | NaN | | NaN | | 1330 | 10 | 9.5103E-14 | 6.3850E+03 |
| 50000 | NaN | | NaN | | 1099 | 8 | 7.0693E-14 | 4.7826E+03 |
| 60000 | NaN | | NaN | | 1091 | 9 | 7.9640E-14 | 7.6655E+03 |
| 70000 | NaN | | NaN | | 1537 | 11 | 6.9828E-14 | 9.5479E+03 |
| 80000 | NaN | | NaN | | 1123 | 9 | 9.3465E-14 | 1.0452E+04 |
| 90000 | NaN | | NaN | | 1187 | 8 | 8.2007E-14 | 6.9342E+03 |

**Table 5.10:** *Results of Algorithm 12, Algorithm 13 and Algorithm 21 on Baheux-type problems when* $\delta = 0.2$

| | ALgorithm 12 | | Algorithm 13 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\sum k$ | | $\|\| r_k \|\|$ | t(sec) |
| 100 | NaN | | NaN | | 350 | 3 | 9.5666E-14 | 9.11144E-01 |
| 500 | NaN | | NaN | | 956 | 7 | 8.8636E-14 | 2.6458E+00 |
| 1000 | NaN | | NaN | | 1144 | 8 | 9.6769E-14 | 3.6110E+00 |
| 5000 | NaN | | NaN | | 883 | 8 | 7.7114E-14 | 9.8678E+01 |
| 10000 | NaN | | NaN | | 1256 | 10 | 7.3386E-14 | 3.1896E+02 |
| 20000 | NaN | | NaN | | 1225 | 9 | 9.3186E-14 | 8.6909E+02 |
| 30000 | NaN | | NaN | | 1322 | 10 | 6.1560E-14 | 2.2937E+03 |
| 40000 | NaN | | NaN | | 1741 | 13 | 7.4743E-14 | 4.0836E+03 |
| 50000 | NaN | | NaN | | 1287 | 10 | 9.7009E-14 | 5.7353E+03 |
| 60000 | NaN | | NaN | | 913 | 6 | 9.2335E-14 | 4.0548E+03 |
| 70000 | NaN | | NaN | | 1108 | 10 | 9.2116E-14 | 9.5479E+04 |
| 80000 | NaN | | NaN | | 1183 | 9 | 9.4617E-14 | 1.1760E+04 |
| 90000 | NaN | | NaN | | 922 | 7 | 9.7095E-14 | 7.1983E+03 |

**Table 5.11:** *Results of Algorithm 12, ALgorithm 13 and Algorithm 21 on Baheux-type problems when δ = 5*

| | Algorithm 12 | | Algorithm 13 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | t(sec) | $\|r_k\|$ | t(sec) | $\sum k$ | | $\| r_k \|$ | t(sec) |
| 100 | NaN | | NaN | | 917 | 5 | 6.4851E-14 | 1.3184E+00 |
| 500 | NaN | | NaN | | 1045 | 5 | 7.6471E-14 | 1.8536E+00 |
| 1000 | NaN | | NaN | | 1188 | 5 | 7.6335E-14 | 2.8236E+00 |
| 5000 | NaN | | NaN | | 1483 | 8 | 8.6355E-14 | 1.6630E+02 |
| 10000 | NaN | | NaN | | 1368 | 8 | 9.7689E-14 | 5.3811E+02 |
| 20000 | NaN | | NaN | | 1626 | 8 | 4.9152E-14 | 1.4922E+03 |
| 30000 | NaN | | NaN | | 1786 | 8 | 9.7725E-14 | 2.5536E+03 |
| 40000 | NaN | | NaN | | 1351 | 7 | 7.9681E-14 | 4.5328E+03 |
| 50000 | NaN | | NaN | | 1836 | 8 | 9.6394E-14 | 8.1866E+03 |
| 60000 | NaN | | NaN | | 1055 | 5 | 7.3147E-14 | 4.6252E+03 |
| 70000 | NaN | | NaN | | 3073 | 12 | 9.0090E-14 | 2.2036E+04 |
| 80000 | NaN | | NaN | | 2405 | 12 | 7.0835E-14 | 2.3165E+04 |
| 90000 | NaN | | NaN | | 1379 | 7 | 9.0026E-14 | 1.5754E+04 |

**Table 5.12:** *Results of Algorithm 12, Algorithm 13 and Algorithm 21 on Baheux-type problems when δ = 8*

| | Algorithm 12 | | Algorithm 13 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | t(sec) | $\|r_k\|$ | t(sec) | $\sum k$ | | $\| r_k \|$ | t(sec) |
| 100 | NaN | | NaN | | 1034 | 6 | 9.8010E-14 | 1.9315E+00 |
| 500 | NaN | | NaN | | 1043 | 7 | 7.7366E-14 | 3.9604E+00 |
| 1000 | NaN | | NaN | | 1031 | 7 | 9.0470E-14 | 5.5949E+00 |
| 5000 | NaN | | NaN | | 1415 | 8 | 9.4337E-14 | 9.4651E+01 |
| 10000 | NaN | | NaN | | 1298 | 8 | 5.0643E-14 | 4.0015E+02 |
| 20000 | NaN | | NaN | | 1504 | 9 | 9.9206E-14 | 1.8766E+03 |
| 30000 | NaN | | NaN | | 1911 | 11 | 3.0963E-14 | 3.2570E+03 |
| 40000 | NaN | | NaN | | 2043 | 11 | 7.2050E-14 | 4.9437E+03 |
| 50000 | NaN | | NaN | | 2716 | 14 | 9.9222E-14 | 1.1118E+04 |
| 60000 | NaN | | NaN | | 2844 | 14 | 5.9696E-14 | 1.7674E+04 |
| 70000 | NaN | | NaN | | 1571 | 9 | 7.4233E-14 | 1.1866E+04 |
| 80000 | NaN | | NaN | | 1781 | 10 | 3.5099E-14 | 1.3797E+04 |
| 90000 | NaN | | NaN | | 2291 | 13 | 7.1423E-14 | 2.7015E+04 |

## 5.5 Switching between Algorithm 16 and Algorithm 18

Here the Algorithm 21 is initially started with either Algorithm 16 or Algorithm 18, and after executing few iterations, it is halted before breakdown and then the switch to either of them chosen randomly is carried out.

### 5.5.1 Numerical Results

The results obtained with Algorithm 11, Algorithm 14 and the switching Algorithm 21 on Baheux-type problems of different dimensions, for different values of $\delta$ are shown in Tables 5.13-5.16.

**Table 5.13:** *Results of Algorithm 11, Algorithm 14 and Algorithm 21 on Baheux-type problems when* $\delta = 0$

| | Algorithm 11 | | Algorithm 14 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | t(sec) | $\|r_k\|$ | t(sec) | $\sum k$ | | $\| r_k \|$ | t(sec) |
| 100 | NaN | | NaN | | 149 | 2 | 5.4429E-14 | 9.9340E-01 |
| 500 | NaN | | NaN | | 727 | 7 | 8.4056E-14 | 3.9006E+00 |
| 1000 | NaN | | NaN | | 639 | 8 | 9.8752E-14 | 3.6964E+00 |
| 5000 | NaN | | NaN | | 615 | 8 | 8.5117E-14 | 7.8303E+01 |
| 10000 | NaN | | NaN | | 836 | 9 | 9.0758E-14 | 2.6847E+02 |
| 20000 | NaN | | NaN | | 927 | 10 | 9.1009E-14 | 8.4619E+02 |
| 30000 | NaN | | NaN | | 1005 | 10 | 8.6259E-14 | 2.2710E+03 |
| 40000 | NaN | | NaN | | 1111 | 11 | 7.5053E-14 | 3.5692E+03 |
| 50000 | NaN | | NaN | | 1022 | 11 | 9.5679E-14 | 5.3113E+03 |
| 60000 | NaN | | NaN | | 1042 | 11 | 9.0590E-14 | 9.1684E+03 |
| 70000 | NaN | | NaN | | 692 | 8 | 8.7268E-14 | 5.3840E+03 |
| 80000 | NaN | | NaN | | 750 | 8 | 9.4198E-14 | 8.7096E+03 |
| 90000 | NaN | | NaN | | 763 | 8 | 8.8979E-14 | 5.6749E+03 |

**Table 5.14:** *Results of Algorithm 11, ALgorithm 14 and Algorithm 21 on Baheux-type problems when* $\delta = 0.2$

| | ALgorithm 11 | | Algorithm 14 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | t(sec) | $\|r_k\|$ | t(sec) | $\sum k$ | | $\| r_k \|$ | t(sec) |
| 100 | NaN | | NaN | | 326 | 3 | 6.8559E-14 | 2.0174E+00 |
| 500 | NaN | | NaN | | 822 | 8 | 8.6396E-14 | 8.0691E+00 |
| 1000 | NaN | | NaN | | 812 | 7 | 9.3881E-14 | 2.0436E+01 |
| 5000 | NaN | | NaN | | 1002 | 10 | 4.9914E-14 | 1.1611E+02 |
| 10000 | NaN | | NaN | | 1018 | 9 | 9.1225E-14 | 4.5456E+02 |
| 20000 | NaN | | NaN | | 964 | 9 | 8.2257E-14 | 1.4541E+03 |
| 30000 | NaN | | NaN | | 1086 | 10 | 4.9708E-14 | 3.3183E+03 |
| 40000 | NaN | | NaN | | 1085 | 10 | 7.1491E-14 | 2.9397E+03 |
| 50000 | NaN | | NaN | | 1148 | 10 | 7.8168E-14 | 4.4537E+03 |
| 60000 | NaN | | NaN | | 1339 | 13 | 8.3200E-14 | 2.3240E+03 |
| 70000 | NaN | | NaN | | 1017 | 9 | 8.1649E-14 | 5.8636E+03 |
| 80000 | NaN | | NaN | | 1231 | 11 | 8.3876E-14 | 8.4180E+03 |
| 90000 | NaN | | NaN | | 844 | 8 | 9.5212E-14 | 5.7511E+03 |

**Table 5.15:** *Results of Algorithm 11, Algorithm 14 and Algorithm 21 on Baheux-type problems when δ = 5*

| | Algorithm 11 | | Algorithm 14 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | t(sec) | $\|r_k\|$ | t(sec) | $\sum k$ | | $\| r_k \|$ | t(sec) |
| 100 | NaN | | NaN | | 807 | 7 | 8.9095E-14 | 1.9904E+00 |
| 500 | NaN | | NaN | | 763 | 7 | 7.8831E-14 | 3.2541E+00 |
| 1000 | NaN | | NaN | | 995 | 8 | 5.1753E-14 | 2.7976E+00 |
| 5000 | NaN | | NaN | | 1132 | 9 | 9.1870E-14 | 8.2868E+01 |
| 10000 | NaN | | NaN | | 1120 | 9 | 6.3418E-14 | 2.6316E+02 |
| 20000 | NaN | | NaN | | 975 | 9 | 6.4810E-14 | 4.3373E+02 |
| 30000 | NaN | | NaN | | 1145 | 10 | 7.8598E-14 | 2.1784E+03 |
| 40000 | NaN | | NaN | | 1180 | 9 | 6.6699E-14 | 3.1753E+03 |
| 50000 | NaN | | NaN | | 1200 | 10 | 8.1069E-14 | 5.7859E+03 |
| 60000 | NaN | | NaN | | 980 | 9 | 7.3003E-14 | 5.9071E+03 |
| 70000 | NaN | | NaN | | 1239 | 10 | 4.9754E-14 | 8.5466E+03 |
| 80000 | NaN | | NaN | | 1226 | 10 | 5.0487E-14 | 9.8900E+03 |
| 90000 | NaN | | NaN | | 1044 | 9 | 6.9211E-14 | 7.6570E+03 |

**Table 5.16:** *Results of Algorithm 11, Algorithm 14 and Algorithm 21 on Baheux-type problems when δ = 8*

| | Algorithm 11 | | Algorithm 14 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | t(sec) | $\|r_k\|$ | t(sec) | $\sum k$ | | $\| r_k \|$ | t(sec) |
| 100 | NaN | | NaN | | 641 | 6 | 8.6998E-14 | 1.2136E+00 |
| 500 | NaN | | NaN | | 906 | 8 | 8.5903E-14 | 2.1504E+00 |
| 1000 | NaN | | NaN | | 936 | 8 | 8.7768E-14 | 1.6959E+00 |
| 5000 | NaN | | NaN | | 1019 | 9 | 8.7385E-14 | 3.9265E+01 |
| 10000 | NaN | | NaN | | 1128 | 10 | 8.0363E-14 | 3.0057E+02 |
| 20000 | NaN | | NaN | | 1114 | 10 | 5.8076E-14 | 8.9572E+02 |
| 30000 | NaN | | NaN | | 1162 | 11 | 9.2923E-14 | 2.7459E+03 |
| 40000 | NaN | | NaN | | 1195 | 11 | 5.5494E-14 | 2.5727E+03 |
| 50000 | NaN | | NaN | | 1520 | 13 | 7.4189E-14 | 4.8640E+03 |
| 60000 | NaN | | NaN | | 1056 | 10 | 9.7923E-14 | 6.4514E+03 |
| 70000 | NaN | | NaN | | 1331 | 13 | 5.4630E-14 | 9.4084E+03 |
| 80000 | NaN | | NaN | | 1285 | 11 | 6.1049E-14 | 8.7633E+03 |
| 90000 | NaN | | NaN | | 1113 | 11 | 9.5249E-14 | 8.4065E+03 |

## 5.6 Switching between Algorithm 16 and Algorithm 19

Algorithm 21 is started with either Algorithm 16 or Algorithm 19, i.e. one of the algorithms run and halted before breakdown and then the switch to either of them chosen randomly is carried out.

### 5.6.1 Numerical Results

The results obtained with Algorithm 11, Algorithm 13 and the switching Algorithm 21 on Baheux-type problems of different dimensions, for different values of $\delta$ are shown in Tables 5.17-5.20.

**Table 5.17:** *Results of Algorithm 11, Algorithm 13 and Algorithm 21 on Baheux-type problems when* $\delta = 0$

| | Algorithm 11 | | Algorithm 13 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | t(sec) | $\|r_k\|$ | t(sec) | $\sum k$ | | $\| r_k \|$ | t(sec) |
| 100 | NaN | | NaN | | 143 | 2 | 6.2489E-14 | 1.2875E+00 |
| 500 | NaN | | NaN | | 845 | 8 | 8.7395E-14 | 6.0516E+00 |
| 1000 | NaN | | NaN | | 1151 | 11 | 9.0760E-14 | 1.5740E+01 |
| 5000 | NaN | | NaN | | 1328 | 13 | 9.1996E-14 | 1.7189E+02 |
| 10000 | NaN | | NaN | | 850 | 8 | 9.9440E-14 | 3.3005E+02 |
| 20000 | NaN | | NaN | | 1156 | 10 | 9.3026E-14 | 1.3018E+03 |
| 30000 | NaN | | NaN | | 1313 | 11 | 8.9071E-14 | 5.9207E+03 |
| 40000 | NaN | | NaN | | 952 | 9 | 7.4415E-14 | 5.2736E+03 |
| 50000 | NaN | | NaN | | 2028 | 19 | 9.6995E-14 | 1.1173E+04 |
| 60000 | NaN | | NaN | | 1454 | 13 | 9.5329E-14 | 1.3430E+04 |
| 70000 | NaN | | NaN | | 1360 | 12 | 7.2751E-14 | 1.0816E+04 |
| 80000 | NaN | | NaN | | 1312 | 12 | 9.9669E-14 | 1.3736E+04 |
| 90000 | NaN | | NaN | | 885 | 8 | 9.5321E-14 | 9.9720E+03 |

**Table 5.18:** *Results of Algorithm 11, Algorithm 13 and Algorithm 21 on Baheux-type problems when* $\delta = 0.2$

| | Algorithm 11 | | Algorithm 13 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | t(sec) | $\|r_k\|$ | t(sec) | $\sum k$ | | $\| r_k \|$ | t(sec) |
| 100 | NaN | | NaN | | 361 | 4 | 8.4116E-14 | 1.4017E+00 |
| 500 | NaN | | NaN | | 1196 | 11 | 8.2264E-14 | 9.9396E+00 |
| 1000 | NaN | | NaN | | 753 | 7 | 8.1432E-14 | 1.2279E+01 |
| 5000 | NaN | | NaN | | 957 | 9 | 8.3569E-14 | 1.8631E+02 |
| 10000 | NaN | | NaN | | 1013 | 10 | 6.3348E-14 | 8.1048E+02 |
| 20000 | NaN | | NaN | | 765 | 7 | 6.7564E-14 | 1.6621E+03 |
| 30000 | NaN | | NaN | | 1121 | 10 | 7.9328E-14 | 4.0938E+03 |
| 40000 | NaN | | NaN | | 1273 | 10 | 9.8905E-14 | 4.1029E+03 |
| 50000 | NaN | | NaN | | 926 | 8 | 8.4312E-14 | 5.1431E+03 |
| 60000 | NaN | | NaN | | 1716 | 14 | 5.4933E-14 | 1.2862E+04 |
| 70000 | NaN | | NaN | | 1132 | 10 | 8.2734E-14 | 9.4392E+03 |
| 80000 | NaN | | NaN | | 1770 | 16 | 8.2071E-14 | 2.3258E+04 |
| 90000 | NaN | | NaN | | 1291 | 12 | 8.2316E-14 | 2.3370E+04 |

**Table 5.19:** *Results of Algorithm 11, Algorithm 13 and Algorithm 21 on Baheux-type problems when δ = 5*

| | Algorithm 11 | | Algorithm 13 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | t(sec) | $\|r_k\|$ | t(sec) | $\sum k$ | | $\| r_k \|$ | t(sec) |
| 100 | NaN | | NaN | | 657 | 5 | 6.1731E-14 | 1.0476E+00 |
| 500 | NaN | | NaN | | 962 | 7 | 9.4535E-14 | 2.9009E+00 |
| 1000 | NaN | | NaN | | 960 | 8 | 7.9350E-14 | 3.5007E+00 |
| 5000 | NaN | | NaN | | 906 | 8 | 7.2284E-14 | 6.1979E+01 |
| 10000 | NaN | | NaN | | 1001 | 9 | 7.8009E-14 | 2.2937E+02 |
| 20000 | NaN | | NaN | | 1160 | 10 | 8.4021E-14 | 8.8056E+02 |
| 30000 | NaN | | NaN | | 1254 | 11 | 6.1894E-14 | 2.1647E+03 |
| 40000 | NaN | | NaN | | 1141 | 10 | 6.9793E-14 | 2.7633E+03 |
| 50000 | NaN | | NaN | | 1047 | 9 | 6.8905E-14 | 6.3576E+03 |
| 60000 | NaN | | NaN | | 1144 | 10 | 9.8689E-14 | 8.4959E+03 |
| 70000 | NaN | | NaN | | 1091 | 9 | 8.5125E-14 | 1.2639E+04 |
| 80000 | NaN | | NaN | | 1233 | 10 | 3.7479E-14 | 1.9849E+04 |
| 90000 | NaN | | NaN | | 915 | 8 | 6.8924E-14 | 1.5634E+04 |

**Table 5.20:** *Results of Algorithm 11, Algorithm 13 and Algorithm 21 on Baheux-type problems when δ = 8*

| | Algorithm 11 | | Algorithm 13 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | t(sec) | $\|r_k\|$ | t(sec) | $\sum k$ | | $\| r_k \|$ | t(sec) |
| 100 | NaN | | NaN | | 896 | 8 | 9.0693E-14 | 2.1977E+00 |
| 500 | NaN | | NaN | | 925 | 9 | 8.7449E-14 | 2.7038E+00 |
| 1000 | NaN | | NaN | | 899 | 9 | 5.4287E-14 | 4.2063E+00 |
| 5000 | NaN | | NaN | | 906 | 9 | 9.9703E-14 | 7.8573E+01 |
| 10000 | NaN | | NaN | | 882 | 9 | 9.7803E-14 | 1.8528E+02 |
| 20000 | NaN | | NaN | | 1134 | 11 | 9.8800E-14 | 1.2316E+03 |
| 30000 | NaN | | NaN | | 1067 | 10 | 8.1048E-14 | 3.3113E+03 |
| 40000 | NaN | | NaN | | 1066 | 10 | 8.3712E-14 | 5.2839E+03 |
| 50000 | NaN | | NaN | | 1322 | 12 | 5.7524E-14 | 7.2849E+03 |
| 60000 | NaN | | NaN | | 902 | 9 | 9.5416E-14 | 6.3544E+03 |
| 70000 | NaN | | NaN | | 1178 | 11 | 4.7848E-14 | 1.3715E+04 |
| 80000 | NaN | | NaN | | 1218 | 11 | 9.0713E-14 | 1.7708E+04 |
| 90000 | NaN | | NaN | | 989 | 10 | 6.1215E-14 | 1.5162E+04 |

## 5.7 Switching between Algorithm 18 and Algorithm 19

Here the Algorithm 21 is initially started with either Algorithm 18 or Algorithm 19, and after executing few iterations, it is halted before breakdown and then the switch to either of them chosen randomly is carried out.

### 5.7.1 Numerical Results

The results obtained with Algorithm 13, Algorithm 14 and the switching Algorithm 21 on Baheux-type problems of different dimensions, for different values of $\delta$ are shown in Tables 5.21-5.24.

**Table 5.21:** *Results of Algorithm 13, Algorithm 14 and Algorithm 21 on Baheux-type problems when $\delta = 0$*

| | Algorithm 13 | | Algorithm 14 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | t(sec) | $\|r_k\|$ | t(sec) | $\sum k$ | | $\| r_k \|$ | t(sec) |
| 100 | NaN | | NaN | | 277 | 4 | 7.1133E-14 | 1.2136E+00 |
| 500 | NaN | | NaN | | 655 | 9 | 7.0303E-14 | 2.3065E+00 |
| 1000 | NaN | | NaN | | 794 | 10 | 7.7291E-14 | 2.4763E+00 |
| 5000 | NaN | | NaN | | 676 | 9 | 8.8608E-14 | 4.9447E+01 |
| 10000 | NaN | | NaN | | 670 | 9 | 9.0541E-14 | 9.3519E+01 |
| 20000 | NaN | | NaN | | 1120 | 13 | 9.4046E-14 | 1.0936E+03 |
| 30000 | NaN | | NaN | | 1003 | 12 | 7.4658E-14 | 2.0624E+03 |
| 40000 | NaN | | NaN | | 1230 | 14 | 9.9142E-14 | 5.0916E+03 |
| 50000 | NaN | | NaN | | 791 | 11 | 9.9051E-14 | 2.4449E+03 |
| 60000 | NaN | | NaN | | 1104 | 14 | 8.0392E-14 | 7.0366E+03 |
| 70000 | NaN | | NaN | | 939 | 12 | 7.0694E-14 | 9.0725E+03 |
| 80000 | NaN | | NaN | | 1220 | 15 | 9.3741E-14 | 1.5110E+04 |
| 90000 | NaN | | NaN | | 1015 | 13 | 7.3478E-14 | 1.4508E+04 |

**Table 5.22:** *Results of Algorithm 13, Algorithm 14 and Algorithm 21 on Baheux-type problems when $\delta = 0.2$*

| | Algorithm 13 | | Algorithm 14 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | t(sec) | $\|r_k\|$ | t(sec) | $\sum k$ | | $\| r_k \|$ | t(sec) |
| 100 | NaN | | NaN | | 413 | 5 | 7.6390E-14 | 1.1341E+00 |
| 500 | NaN | | NaN | | 1044 | 11 | 5.8548E-14 | 1.7227E+00 |
| 1000 | NaN | | NaN | | 1248 | 13 | 9.5118E-14 | 4.6651E+00 |
| 5000 | NaN | | NaN | | 1295 | 14 | 8.5197E-14 | 6.1776E+01 |
| 10000 | NaN | | NaN | | 1252 | 13 | 8.6696E-14 | 3.0083E+02 |
| 20000 | NaN | | NaN | | 1000 | 11 | 8.2636E-14 | 4.5507E+02 |
| 30000 | NaN | | NaN | | 1115 | 12 | 9.7625E-14 | 1.6191E+03 |
| 40000 | NaN | | NaN | | 881 | 9 | 9.6819E-14 | 2.2483E+03 |
| 50000 | NaN | | NaN | | 1200 | 13 | 9.4498E-14 | 4.4876E+03 |
| 60000 | NaN | | NaN | | 1211 | 13 | 7.1510E-14 | 6.4184E+03 |
| 70000 | NaN | | NaN | | 1110 | 12 | 9.4697E-14 | 8.3229E+03 |
| 80000 | NaN | | NaN | | 1029 | 11 | 6.7186E-14 | 1.0473E+04 |
| 90000 | NaN | | NaN | | 1144 | 12 | 8.2802E-14 | 1.6649E+04 |

**Table 5.23:** *Results of Algorithm 13, Algorithm 14 and Algorithm 21 on Baheux-type problems when δ = 5*

| | Algorithm 13 | | Algorithm 14 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | t(sec) | $\|r_k\|$ | t(sec) | $\sum k$ | | $\| r_k \|$ | t(sec) |
| 100 | NaN | | NaN | | 642 | 5 | 7.3332E-14 | 1.2698E+00 |
| 500 | NaN | | NaN | | 783 | 6 | 9.9579E-14 | 1.3623E+00 |
| 1000 | NaN | | NaN | | 1085 | 8 | 9.2735E-14 | 2.7276E+00 |
| 5000 | NaN | | NaN | | 1209 | 9 | 9.2111E-14 | 6.0750E+01 |
| 10000 | NaN | | NaN | | 1205 | 9 | 8.8425E-14 | 1.4547E+02 |
| 20000 | NaN | | NaN | | 1368 | 10 | 6.8308E-14 | 9.1618E+02 |
| 30000 | NaN | | NaN | | 1517 | 11 | 7.7128E-14 | 3.1839E+03 |
| 40000 | NaN | | NaN | | 1941 | 14 | 8.6248E-14 | 7.8021E+03 |
| 50000 | NaN | | NaN | | 1372 | 10 | 8.7284E-14 | 6.7612E+03 |
| 60000 | NaN | | NaN | | 1523 | 11 | 3.8548E-14 | 9.5253E+03 |
| 70000 | NaN | | NaN | | 1369 | 10 | 6.3259E-14 | 1.1009E+04 |
| 80000 | NaN | | NaN | | 1521 | 11 | 8.8217E-14 | 1.2977E+04 |
| 90000 | NaN | | NaN | | 1662 | 12 | 6.5920E-14 | 2.3064E+04 |

**Table 5.24:** *Results of Algorithm 13, Algorithm 14 and Algorithm 21 on Baheux-type problems when δ = 8*

| | Algorithm 13 | | Algorithm 14 | | Algorithm 21 | | | |
|---|---|---|---|---|---|---|---|---|
| Prob. size | | | | | Total-numit | Cycles | Residual Norm | Elapsed time |
| $n_1 \times n_2 = n$ | $\|r_k\|$ | t(sec) | $\|r_k\|$ | t(sec) | $\sum k$ | | $\| r_k \|$ | t(sec) |
| 100 | NaN | | NaN | | 693 | 6 | 7.7736E-14 | 1.2808E+00 |
| 500 | NaN | | NaN | | 937 | 8 | 9.2642E-14 | 1.7363E+00 |
| 1000 | NaN | | NaN | | 1066 | 9 | 7.9663E-14 | 4.2646E+00 |
| 5000 | NaN | | NaN | | 1208 | 10 | 7.2772E-14 | 7.6251E+01 |
| 10000 | NaN | | NaN | | 1195 | 10 | 6.7539E-14 | 2.8151E+02 |
| 20000 | NaN | | NaN | | 1195 | 10 | 8.8780E-14 | 1.4111E+03 |
| 30000 | NaN | | NaN | | 1557 | 13 | 7.5170E-14 | 3.6014E+03 |
| 40000 | NaN | | NaN | | 1457 | 12 | 3.7015E-14 | 5.2550E+03 |
| 50000 | NaN | | NaN | | 1567 | 13 | 9.0659E-14 | 8.1525E+03 |
| 60000 | NaN | | NaN | | 1690 | 14 | 6.6446E-14 | 1.3709E+04 |
| 70000 | NaN | | NaN | | 1448 | 12 | 7.2670E-14 | 1.1283E+04 |
| 80000 | NaN | | NaN | | 1317 | 11 | 7.8829E-14 | 1.3046E+04 |
| 90000 | NaN | | NaN | | 1341 | 11 | 8.1043E-14 | 1.3359E+04 |

## 5.8 Comparison between restarting and switching strategies

It can be observed from the results that the proposed switching algorithms are faster than the restarting ones especially when the problems are of high dimensions. As can be seen in Tables 5.25-5.28, these algorithms appear to have the same performance in terms of accuracy.

### 5.8.1 Comparing Algorithm 20 with Algorithm 21, based on $A_4$ and $A_{12}$

The results obtained from Algorithm 20, which run separately for Algorithm 17 and Algorithm 16, are based on relation $A_4$ and $A_{12}$ respectively, are compared with the result of Algorithm 21. The Algorithm 21 is a switching algorithm between Algorithm 17 and Algorithm 16. Numerical results for different values of $\delta = 0$ and $\delta = 0.2$ are recorded in the following Tables 5.25-5.26.

**Table 5.25:** *A comparison of the restarting algorithms, Algorithm 17 and Algorithm 16 against the switching algorithm, Algorithm 21 on a Baheux-type problems of different sizes when* $\delta = 0$

| Dim of Prob | Algorithm 17 | | Algorithm 16 | | Algorithm 21 | |
|---|---|---|---|---|---|---|
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) | $\|\|r_k\|\|$ | t(sec) |
| 100 | 4.9751E-14 | 7.0910E-01 | 5.4429E-14 | 7.5120E-01 | 6.4799E-14 | 1.0229E+00 |
| 500 | 9.7847E-14 | 1.9640E+00 | 9.8779E-14 | 1.8921E+00 | 9.9621E-14 | 1.3673E+00 |
| 1000 | 9.1003E-14 | 4.7824E+00 | 7.8942E-14 | 1.8649E+00 | 8.3930E-14 | 1.5031E+00 |
| 5000 | 9.3487E-14 | 3.6608E+01 | 9.6411E-14 | 6.2534E+01 | 9.3487E-14 | 3.0330E+01 |
| 10000 | 9.9417E-14 | 1.5808E+02 | 8.6591E-14 | 1.2378E+02 | 9.2032E-14 | 1.3467E+02 |
| 20000 | 9.9324E-14 | 6.9171E+02 | 9.0168E-14 | 1.2158E+03 | 8.8114E-14 | 3.7645E+02 |
| 30000 | 9.9248E-14 | 3.4193E+03 | 6.2128E-14 | 1.7086E+03 | 9.5415E-14 | 8.7870E+02 |
| 40000 | 7.5591E-14 | 2.5580E+03 | 8.5319E-14 | 2.9172E+03 | 9.8820E-14 | 1.5565E+03 |
| 50000 | 8.1885E-14 | 2.9318E+03 | 8.8686E-14 | 5.6647E+03 | 9.9651E-14 | 2.6257E+03 |
| 60000 | 8.4811E-14 | 7.2413E+03 | 9.6952E-14 | 7.0835E+03 | 9.4797E-14 | 2.3296E+03 |
| 70000 | 8.7667E-14 | 7.3412E+03 | 9.9118E-14 | 9.3068E+03 | 8.0121E-14 | 3.9998E+03 |
| 80000 | 9.9146E-14 | 6.5786E+03 | 9.7447E-14 | 6.9428E+03 | 8.0435E-14 | 5.7842E+03 |
| 90000 | 9.0707E-14 | 5.0874E+03 | 9.3677E-14 | 8.8362E+03 | 6.3203E-14 | 4.8621E+03 |

**Table 5.26:** *A comparison of the restarting algorithms, Algorithm 17 and Algorithm 16 against the switching algorithm, Algorithm 21 on a Baheux-type problems of different sizes, when* $\delta = 0.2$

| Dim of Prob | Algorithm 17 | | Algorithm 16 | | Algorithm 21 | |
|---|---|---|---|---|---|---|
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | sec | $\|\|r_k\|\|$ | sec | $\|\|r_k\|\|$ | sec |
| 100 | 3.4466E-14 | 9.0114E-01 | 8.4201E-14 | 9.3534E-01 | 7.2018E-14 | 1.7407E+00 |
| 500 | 8.6592E-14 | 1.1456E+00 | 8.2836E-14 | 1.6902E+00 | 6.3987E-14 | 1.1266E+00 |
| 1000 | 8.1958E-14 | 2.2509E+00 | 9.6689E-14 | 2.8613E+00 | 8.2681E-14 | 2.0396E+00 |
| 5000 | 9.4303E-14 | 2.9377E+01 | 8.7238E-14 | 4.1919E+01 | 9.3962E-14 | 2.4518E+01 |
| 10000 | 8.9325E-14 | 1.7968E+02 | 9.3045E-14 | 1.2608E+02 | 9.3596E-14 | 8.1512E+01 |
| 20000 | 8.3356E-14 | 3.9526E+02 | 9.3119E-14 | 7.3278E+02 | 8.6842E-14 | 7.7655E+02 |
| 30000 | 8.1079E-14 | 1.3205E+03 | 8.6376E-14 | 2.0646E+03 | 9.2112E-14 | 1.1766E+03 |
| 40000 | 8.9458E-14 | 3.3931E+03 | 7.7838E-14 | 2.9067E+03 | 8.4260E-14 | 2.7420E+03 |
| 50000 | 7.5661E-14 | 3.6229E+03 | 7.8088E-14 | 3.8596E+03 | 9.5880E-14 | 2.6978E+03 |
| 60000 | 9.1927E-14 | 4.7638E+03 | 9.7165E-14 | 8.0922E+03 | 9.6912E-14 | 2.8531E+03 |
| 70000 | 8.2881E-14 | 6.3346E+03 | 9.1179E-14 | 8.8500E+03 | 9.6283E-14 | 3.5907E+03 |
| 80000 | 7.2007E-14 | 5.3332E+03 | 9.4984E-14 | 9.5485E+03 | 9.5650E-14 | 6.0355E+03 |
| 90000 | 8.7481E-14 | 4.4072E+03 | 8.4068E-14 | 1.1762E+04 | 8.3997E-14 | 5.0977E+03 |

### 5.8.2 Comparing Algorithm 20 with Algorithm 21 based on $A_8/B_6$ and $A_8/B_{10}$

Now, we compare the results from Algorithm 20 which run separately Algorithm 18 and Algorithm 19, are based on relations $A_8/B_6$ and $A_8/B_{10}$ respectively, against the Algorithm 21 which switches between Algorithms 18 and 19. Numerical results for different values of $\delta = 0$ and $\delta = 0.2$ are recorded in the following Tables 5.27-5.28.

**Table 5.27:** *A comparison of the restarting algorithms, Algorithm 18 and Algorithm 19 against the switching algorithm, Algorithm 21 on a Baheux-type problems of different sizes, when* $\delta = 0$

| Dim of Prob | Algorithm 18 | | Algorithm 19 | | Algorithm 21 | |
|---|---|---|---|---|---|---|
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | sec | $\|\|r_k\|\|$ | sec | $\|\|r_k\|\|$ | sec |
| 100 | 6.1115E-14 | 6.2013E-01 | 9.3606E-14 | 8.9790E-01 | 7.1133E-14 | 1.2136E+00 |
| 500 | 9.8912E-14 | 5.3377E+00 | 8.8988E-14 | 5.1838E+00 | 7.0303E-14 | 2.3065E+00 |
| 1000 | 8.5325E-14 | 2.8432E+00 | 9.3373E-14 | 2.5808E+01 | 6.3383E-14 | 2.4763E+00 |
| 5000 | 9.0619E-14 | 1.4516E+01 | 9.3051E-14 | 4.6781E+02 | 8.8608E-14 | 4.0568E+01 |
| 10000 | 8.7698E-14 | 6.3695E+01 | 9.1131E-14 | 1.1683E+03 | 9.9832E-14 | 1.4636E+02 |
| 20000 | 9.6487E-14 | 3.2046E+02 | 8.7733E-14 | 1.9221E+03 | 9.4046E-14 | 1.0936E+03 |
| 30000 | 9.7053E-14 | 4.6949E+02 | 7.9795E-14 | 4.0962E+03 | 7.4658E-14 | 2.0624E+03 |
| 40000 | 9.7491E-14 | 9.9272E+02 | 9.5148E-14 | 1.2473E+04 | 9.9142E-14 | 5.0916E+03 |
| 50000 | 8.7656E-14 | 1.2205E+03 | 9.9344E-14 | 1.8289E+03 | 5.5027E-14 | 6.0572E+03 |
| 60000 | 8.7424E-14 | 1.5603E+03 | 9.5721E-14 | 2.8230E+04 | 8.0392E-14 | 7.0366E+03 |
| 70000 | 9.0205E-14 | 1.8936E+03 | 9.1219E-14 | 6.4745E+04 | 7.0694E-14 | 9.0725E+03 |
| 80000 | 9.8981E-14 | 2.7773E+03 | 8.9973E-14 | 3.7393E+04 | 9.3741E-14 | 1.5110E+04 |
| 90000 | 8.4513E-14 | 4.3607E+03 | 9.0799E-14 | 5.8112E+04 | 7.3478E-14 | 1.4508E+04 |

**Table 5.28:** *A comparison of the restarting algorithms, Algorithm 18 and Algorithm 19 against the switching algorithm, Algorithm 21 on a Baheux-type problems of different sizes, when* $\delta = 0.2$

| Dim of Prob | Algorithm 18 | | Algorithm 19 | | Algorithm 21 | |
|---|---|---|---|---|---|---|
| $n_1 \times n_2 = n$ | $\|\|r_k\|\|$ | sec | $\|\|r_k\|\|$ | sec | $\|\|r_k\|\|$ | sec |
| 100 | 7.7436E-14 | 8.4320E-01 | 9.3360E-14 | 9.8385E-01 | 7.6390E-14 | 1.1341E+00 |
| 500 | 5.9065E-14 | 1.5156E+00 | 9.6024E-14 | 1.3427E+01 | 5.8548E-14 | 1.7227E+00 |
| 1000 | 7.9484E-14 | 2.4188E+00 | 8.2166E-14 | 1.9866E+01 | 9.5118E-14 | 4.6651E+00 |
| 5000 | 9.7027E-14 | 7.6450E+01 | 8.7901E-14 | 1.2460E+02 | 8.5197E-14 | 6.1776E+01 |
| 10000 | 8.6415E-14 | 2.1527E+02 | 9.9181E-14 | 8.3228E+02 | 8.6696E-14 | 3.0083E+02 |
| 20000 | 5.8727E-14 | 4.2976E+02 | 9.5309E-14 | 2.4773E+03 | 8.2636E-14 | 4.5507E+02 |
| 30000 | 9.6493E-14 | 1.7039E+03 | 7.9825E-14 | 1.0237E+04 | 9.7625E-14 | 1.6191E+03 |
| 40000 | 8.3941E-14 | 2.5913E+03 | 8.9642E-14 | 1.4145E+04 | 9.6819E-14 | 2.2483E+03 |
| 50000 | 7.4461E-14 | 3.9558E+03 | 8.1724E-14 | 1.1617E+04 | 9.4498E-14 | 4.4876E+03 |
| 60000 | 8.6231E-14 | 4.7694E+03 | 9.9193E-14 | 1.6511E+04 | 7.1510E-14 | 6.4184E+03 |
| 70000 | 7.7883E-14 | 5.7245E+03 | 8.4839E-14 | 4.0159E+04 | 9.4697E-14 | 8.3229E+03 |
| 80000 | 7.9999E-14 | 8.5143E+03 | 8.9733E-14 | 4.2025E+04 | 6.7186E-14 | 1.0473E+04 |
| 90000 | 9.2184E-14 | 1.4880E+04 | 8.7492E-14 | 2.3436E+04 | 8.2802E-14 | 1.6649E+04 |

## 5.9 Summary

Algorithms $A_4$, $A_{12}$, $A_8/B_6$ and $A_8/B_{10}$ are implemented to solve various problems of the type given in Sections 4.6, 4.7, 4.8 and 4.9, respectively, with different dimensions ranging from 100 to 90000. The results from these algorithms are compared with those from the switching algorithms, i.e. Algorithms 19 to 24 on the same problems. The results reveal that $A_4$, $A_{12}$, $A_8/B_6$ and $A_8/B_{10}$ are not as robust as the switching algorithms. Individual algorithms customarily solved problems with dimension $n \leq 20$ achieving poor accuracy. On the contrary, the switching algorithms solved these problems with a higher accuracy. This argument is supported by strong numerical evidence in favour of switching. It is obvious from the results obtained that switching is an effective strategy to handle the issue of breakdown in Lanczos-type algorithms. It is evident to say that switching strategies can be recommended for efficiency enhancement of the Lanczos-type algorithms along with their robustness. These strategies are also attractive for their simplicity and ease of implementation.

# Chapter 6

# Conclusion and Further Work

This thesis focuses on some iterative methods for solving linear systems of equations (SLEs). These methods are commonly known as Lanczos-type algorithms. Although these algorithms are known for their efficiency, they suffer from a major problem which is that of premature breakdown. This breakdown usually occurs well before convergence to a good approximate solution. This is due to the loss of orthogonality of the Formal Orthogonal Polynomials (FOPs) on which these algorithms are based, due to non-existence of FOPs, accumulation of errors or numerical difficulties while estimating their coefficients. The numerical difficulties in estimating the coefficients occur when these involve denominators which become zero during the computational process.

A number of attempts have been made to deal with the breakdown issue in Lanczos-type algorithms. Some of these attempts provided the foundation for look-ahead algorithms and look-around algorithms [11,18,19,32,40]. Some have led to jumping over non-existing FOPs [23], whereas others have inspired restarting from different points for desirable results in Krylov subspaces [35]. Some of the strategies have considered switching between algorithms to provide a remedy to the breakdown and continue the process until achieving convergence [36]. It has been established that restarting and switching strategies are better than others in terms of robustness [35,36]. However, these strategies have not been applied to problems with large sizes. This work considers substantially larger instances of SLEs than those reported in the available literature. Our results on the whole support our hypothesis on switching and restarting.

In chapter 4, we have advocated the restarting of algorithms before they broke down. A test to detect the forthcoming breakdown is described. It relies on some parameters including the iteration number.

After explaining thoroughly the breakdown issue and some of the existing strategies to handle the breakdown of the Lanczos-type algorithm, a search has also been made to find algorithms that are more robust to the issue of breakdown. This is done by extending the degree of FOPs used in Lanczos-type algorithms. The extended degrees FOPs based algorithms are compared with the existing ones that are based on low degree FOPs. It has, however, been observed that the Lanczos-type algorithms based on high degree FOPs are computationally more expensive than the others. Moreover they face a breakdown issue due to error accumulation at a higher speed than the others.

Furthermore, other variants of Lanczos-type algorithms involving ordinary polynomial and monic polynomial have also been derived instead of standard auxiliary polynomial as used in some previous works by other researchers hinted to in Chapter 3 of the thesis. Further exploration of these algorithms might help in providing more insight into the matter.

The components of those coefficients with the denominators that blow up prior to breakdown are regularly monitored. We have suggested a stopping test based on the value of those components that become less than a specified threshold. This test helps to stop the algorithms preemptively just before breakdown. This allows the algorithms to run for a maximum number of iterations unlike the conventional methods where the algorithms are run for a pre-decided number of iterations. The results given in this thesis have revealed that by utilising the maximum number of iterations, robustness can be achieved. This test is incorporated in both the restarting and switching strategies. The results show that these approaches are good competitors in terms of both their robustness and efficiency in comparison to other conventional methods. Some convergence analysis carried out on well known algorithm is included as appendix A.

## 6.1 Further research work

The generalisation of switching to a whole library of Lanczos-type algorithms may prove very beneficial since it is difficult to match a given Lanczos-type algorithm to a given problem. Here we have considered two-way switching between two distinct algorithms. A worthwhile investigation might be a $k$-way switching or switching between $k$ distinct algorithms. Non-Lanczos-type algorithms might also be considered for this purpose.

This could be done by considering any number of algorithms which are suitable for solving SLEs and switch between them as soon as the current algorithm threatens to breakdown. While hitting on a good algorithm, switching away from it to another algorithm may be counter-productive. It is, therefore, also worthwhile to investigate a combination of switching and restarting. Here, restarting is equivalent to switching to the same algorithm. This can happen when the current algorithm is very appropriate for the SLE instance being solved. "Appropriateness" may be characterised by the number of iterations the algorithm takes before monitoring shows that it is going to breakdown. There is also the analysis of all these approaches in terms of robustness and efficiency.

# Bibliography

[1] Grégoire Allaire, Karim Trabelsi, and Sidi Mahmoud Kaber. *Numerical Linear Algebra*. Springer, 2008.

[2] Roberto Bagnara. A unified proof for the convergence of Jacobi and Gauss-Seidel methods. *SIAM review*, 37(1):93–97, 1995.

[3] Carole Baheux. New implementations of Lanczos method. *Journal of computational and applied mathematics*, 57(1):3–15, 1995.

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

[5] Claude Brezinski. Padé-type approximants. In *Padé-Type Approximation and General Orthogonal Polynomials*, pages 9–39. Springer, 1980.

[6] Claude Brezinski. Projection methods for linear systems. *Journal of Computational and Applied Mathematics*, 77(1-2):35–51, 1997.

[7] Claude Brezinski. A transpose-free "Lanczos/Orthodir" algorithm for linear systems. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 324(3):349–354, 1997.

[8] Claude Brezinski. *Computational aspects of linear control*. Springer Science & Business Media, 2002.

[9] Claude Brezinski. An intoduction to formal orthogonality and some of its applications. *RACSAM*, 96(2):157–173, 2002.

[10] Claude Brezinski, Ufr Ieea, and J Van Iseghem. A taste of Padé approximation. *Acta numerica*, 4:53–103, 1995.

[11] Claude Brezinski and M Redivo-Zaglia. A new presentation of orthogonal polynomials with applications to their computation. *Numerical Algorithms*, 1(2):207–221, 1991.

[12] Claude Brezinski and M Redivo-Zaglia. Breakdowns in the computation of orthogonal polynomials. *Nonlinear Numerical Methods and Rational Approximation, Kluwer, Dordrecht*, pages 49–59, 1994.

[13] Claude Brezinski and M Redivo-Zaglia. Treatment of near-breakdown in the CGS algorithm. *Numerical Algorithms*, 7(1):33–73, 1994.

[14] Claude Brezinski and M Redivo-Zaglia. A look-ahead strategy for the implementation of some old and new extrapolation methods. *Numerical Algorithms*, 11(1):35–55, 1996.

[15] Claude Brezinski, M Redivo-Zaglia, and Hassane Sadok. Avoiding breakdown and near-breakdown in Lanczos–type algorithms. *Numerical Algorithms*, 1(2):261–284, 1991.

[16] Claude Brezinski, M Redivo-Zaglia, and Hassane Sadok. Addendum to "Avoiding breakdown and near-breakdown in Lanczos–type algorithms". *Numerical Algorithms*, 2(2):133–136, 1992.

[17] Claude Brezinski, M Redivo-Zaglia, and Hassane Sadok. A breakdown-free Lanczos–type algorithm for solving linear systems. *Numerische Mathematik*, 63(1):29–38, 1992.

[18] Claude Brezinski, M Redivo-Zaglia, and Hassane Sadok. Breakdowns in the implementation of the Lanczos method for solving linear systems. *Computers & Mathematics with Applications*, 33(1):31–44, 1997.

[19] Claude Brezinski, M Redivo-Zaglia, and Hassane Sadok. New look-ahead Lanczos-type algorithms for linear systems. *Numerische Mathematik*, 83(1):53–85, 1999.

[20] Claude Brezinski, M Redivo-Zaglia, and Hassane Sadok. The matrix and polynomial approaches to Lanczos-type algorithms. *Journal of computational and applied mathematics*, 123(1):241–260, 2000.

[21] Claude Brezinski, M Redivo-Zaglia, and Hassane Sadok. A review of formal orthogonality in Lanczos-based methods. *Journal of Computational and Applied Mathematics*, 140(1):81–98, 2002.

[22] Claude Brezinski and Hassane Sadok. Avoiding breakdown in the CGS algorithm. *Numerical Algorithms*, 1(2):199–206, 1991.

[23] Claude Brezinski and Hassane Sadok. Lanczos-type algorithms for solving systems of linear equation. *Applied numerical mathematics*, 11(6):443–473, 1993.

[24] Daniela Calvetti, L Reichel, and Danny Chris Sorensen. An implicitly restarted Lanczos method for large symmetric eigenvalue problems. *Electronic Transactions on Numerical Analysis*, 2(1):21, 1994.

[25] Zhi-Hao Cao. Avoiding breakdown in variants of the BI-CGSTAB algorithm. *Linear algebra and its applications*, 263:113–132, 1997.

[26] Ke Chen. *Matrix preconditioning techniques and applications*. Cambridge University Press, 2005.

[27] Biswa Nath Datta. *Numerical linear algebra and applications*. SIAM, 2010.

[28] James W Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.

[29] André Draux. Formal orthogonal polynomials and Pade approximants in a non-commutative algebra. In *Mathematical Theory of Networks and Systems*, pages 278–292. Springer, 1984.

[30] André Draux. Formal orthogonal polynomials revisited. Applications. *Numerical Algorithms*, 11(1):143–158, 1996.

[31] André Draux. Formal orthogonal polynomials and Newton–Padé approximants. *Numerical Algorithms*, 29(1-3):67–74, 2002.

[32] Antonio J Durán and Walter Van Assche. Orthogonal matrix polynomials and higher-order recurrence relations. *Linear Algebra and its Applications*, 219:261–280, 1995.

[33] Muhammad Farooq. *New Lanczos-type algorithms and their implementation*. PhD thesis, The University of Essex, 2011.

[34] Muhammad Farooq and Abdellah Salhi. New recurrence relationships between orthogonal polynomials which lead to new Lanczos-type algorithms. *Journal of Prime Research in Mathematics*, 8:61–75, 2012.

[35] Muhammad Farooq and Abdellah Salhi. A preemptive restarting approach to beating the inherent instability of Lanczos-type algorithms. *Iranian Journal of Science and Technology (Sciences)*, 37(3.1):349–358, 2013.

[36] Muhammad Farooq and Abdellah Salhi. A switching approach to avoid breakdown in Lanczos-type algorithms. *Applied Mathematics and Information Sciences*, 8(5):2161–2169, 2014.

[37] Roger Fletcher. Conjugate gradient methods for indefinite systems. *Lecture Notes in Mathematics*, 506:73–89, 1976.

[38] Roland W Freund, Martin H Gutknecht, and Noël M Nachtigal. An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices. *SIAM Journal on Scientific Computing*, 14(1):137–158, 1993.

[39] Walter Gander, Martin J Gander, and Felix Kwok. Scientific Computing: An Introduction using Maple and MATLAB. *Texts in Computational Science and Engineering*, 11, 2014.

[40] Peter R Graves-Morris. A "look-around Lanczos" algorithm for solving a system of linear equations. *Numerical Algorithms*, 15(3):247–274, 1997.

[41] Anne Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, 1997.

[42] Martin H Gutknecht. The unsymmetric Lanczos algorithms and their relations to Padé approximation, continued fractions and the QD algorithm. In *Proceedings of the Copper Mountain Conference on Iterative Methods*, volume 2, 1990.

[43] Martin H Gutknecht. Lanczos-type solvers for nonsymmetric linear systems of equations. *Acta Numerica*, 6:271–397, 1997.

[44] Magnus R Hestenes. The conjugate gradient method for solving linear systems. In *Proc. Symp. Appl. Math VI, American Mathematical Society*, pages 83–102, 1956.

[45] Magnus R Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems'. *Journal of Research of the National Bureau of Standards*, 49(6), 1952.

[46] Desmond J Higham and Nicholas J Higham. *MATLAB guide*. SIAM, 2016.

[47] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

[48] Kang C Jea and David M Young. On the simplification of generalized conjugate-gradient methods for nonsymmetrizable linear systems. *Linear Algebra and its Applications*, 52:399–417, 1983.

[49] Wayne Joubert. Generalized conjugate gradient and Lanczos methods for the solution of nonsymmetric systems of linear equations. Technical report, Texas Univ., Austin, TX (USA). Center for Numerical Analysis, 1990.

[50] Wayne Joubert. Lanczos methods for the solution of nonsymmetric systems of linear equations. *SIAM Journal on Matrix Analysis and Applications*, 13(3):926–943, 1992.

[51] Louis Komzsik. *The Lanczos method: evolution and application*. SIAM, 2003.

[52] Cornelius Lanczos. *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*. United States Governm. Press Office, 1950.

[53] Cornelius Lanczos. Solution of systems of linear equations by minimized iterations. *J. Res. Nat. Bur. Standards*, 49(1):33–53, 1952.

[54] Jörg Liesen and Zdenek Strakos. *Krylov subspace methods: principles and analysis*. Oxford University Press, 2013.

[55] Maharani Maharani. *Enhanced Lanczos Algorithms for Solving Systems of Linear Equations with Embedding Interpolation and Extrapolation*. PhD thesis, University of Essex, 2015.

[56] Gérard Meurant and Zdeněk Strakoš. The Lanczos and conjugate gradient algorithms in finite precision arithmetic. *Acta Numerica*, 15:471–542, 2006.

[57] Ronald Morgan. On restarting the Arnoldi method for large nonsymmetric eigenvalue problems. *Mathematics of Computation of the American Mathematical Society*, 65(215):1213–1230, 1996.

[58] Dywayne A Nicely. *Restarting the Lanczos algorithm for large eigenvalue problems and linear equations*. Baylor University, 2008.

[59] Christopher C Paige. *The computation of eigenvalues and eigenvectors of very large sparse matrices.* PhD thesis, University of London, 1971.

[60] Beresford N Parlett, Derek R Taylor, and Zhishun A Liu. A look-ahead Lanczos algorithm for unsymmetric matrices. *Mathematics of Computation*, 44(169):105–124, 1985.

[61] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical Mathematics*. Springer, 2007.

[62] Yousef Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Mathematics of computation*, 37(155):105–126, 1981.

[63] Yousef Saad. On the Lanczos method for solving symmetric linear systems with several right-hand sides. *Mathematics of computation*, 48(178):651–662, 1987.

[64] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[65] Yousef Saad. *Numerical Methods for Large Eigenvalue Problems: Revised Edition*. SIAM, 2011.

[66] Peter Sonneveld. CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 10(1):36–52, 1989.

[67] Gábor Szegö. *Orthogonal polynomials*. American Mathematical Soc., 1939.

[68] Henk A Van der Vorst. An iterative solution method for solving f(A)x=b, using Krylov subspace information obtained for the symmetric positive definite matrix A. *Journal of Computational and Applied Mathematics*, 18(2):249–263, 1987.

[69] Henk A Van der Vorst. *Iterative Krylov methods for large linear systems*. Cambridge University Press, 2003.

[70] Richard S Varga. *Matrix iterative analysis*. Springer, 2009.

[71] Jet Wimp. Padé-type approximation and general orthogonal polynomials. *SIAM Review*, 23(3):403–406, 1981.

# Publications / Talks

1. I presented part my work at EGH2015 held at the Department of Mathematical Sciences, University of Essex (June 3, 2015).

# Appendix A

# Basic and Auxiliary Results

## A.1 Convergence Analysis of Iterative Methods

In general, consider an iterative solution of an $n \times n$ system of linear equation $Ax = b$ as

$$x_{m+1} = Bx_m + c, \tag{A.1.1}$$

$$x_{m+1} = x_m + M^{-1}r_m \tag{A.1.2}$$

where $r_m = b - Ax_m$ denotes the residual vector at step $m$, the matrix $B$ is called an iteration matrix $B = M^{-1}N$, while $c = M^{-1}b$.

Suppose the sequence $\{x_m\}_{m=0}^{\infty}$ has to converges to the exact solution $x$. Since the error has the form

$$e_{m+1} = Be_m,$$

by induction on $m$, we obtain

$$e_m = B^m e_0, \tag{A.1.3}$$

where $e_0$ is the initial error. Taking the norm on both sides, then

$$||e_m|| = ||B^m e_0|| \Rightarrow ||e_m|| \leq ||B^m|| ||e_0|| = ||B||^m ||e_0||.$$

If $||B|| < 1$, then $||B||^m \to 0$ as $m \to \infty$ and hence, $x_m \to x$ as $m \to \infty$ [26,70]. [2]

To carry out the convergence analysis of Lanczos/Orthodir and Lanczos/Orthomin algorithms, we follow the same procedure for CG method given in [1,61].

## A.1.1 Convergence analysis of Lanczos/Orthodir

Lanczos/Orthodir algorithm is also called algorithm $A_8/B_6$ in C. Baheux [4]. Using the three-term recurrence relationship we obtain the following expression for the residual and the next solution [3,4].

$$r_{m+1} = r_m + A_{m+1}Az_m \tag{A.1.4}$$

$$x_{m+1} = x_m - A_{m+1}z_m \tag{A.1.5}$$

Since $z_m = P_m^{(1)}(A)r_0$, now subtracting $x^{(*)}$ on both sides of (4.5)

$$e_{m+1} = e_m - A_{m+1}z_m,$$

$$\begin{cases} e_m = e_{m-1} - A_m z_{m-1} \\ e_{m-1} = e_{m-2} - A_{m-1}z_{m-2} \\ e_{m-2} = e_{m-3} - A_{m-2}z_{m-3} \\ \vdots \\ e_3 = e_2 - A_3 z_2 \\ e_2 = e_1 - A_2 z_1 \\ e_1 = e_0 - A_1 z_0 \end{cases}$$

$$e_{m+1} = e_0 - A_1z_0 - A_2z_1 - A_3z_2 - ... - A_{m+1}z_m$$

$$e_{m+1} = e_0 - \sum_{i=0}^{m} A_{i+1}z_i$$

$$||x_{m+1} - x^*|| \leq ||x_0 - x^*|| + \sum_{i=0}^{m} |A_{i+1}|||z_i||$$

$$||x_{m+1} - x^*|| \leq ||x_0 - x^*|| + |A_{m+1}|||z_m||. \tag{A.1.6}$$

Now consider the second part of the equation (A.1.6) on the right hand side

$$||z_m|| = ||P_m^{(1)}(A)r_0||$$

$$||z_m|| \leq ||P_m^{(1)}(A)||||r_0||,$$

$$||x_{m+1} - x^*|| \leq ||x_0 - x^*|| + |A_{m+1}|||P_m^{(1)}(A)||||r_0||.$$

$||.||$ is induced norm. Since $A$ is symmetric positive definite, there exists an orthogonal matrix $V$ such that $A = V\Lambda V^{-1}$ with $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_m)$, i.e. $\lambda_i \in \rho(A)$, where $\rho(A)$ in known as the spectral radius of $A$. Let $\lambda$ be any eigenvalue of matrix $A$. We also assume that the initial guess is chosen such that it is close to the true solution, so that $||r_0|| \leq \epsilon_2$, for some $\epsilon_2 > 0$, we obtain the following result as

$$||z_m|| \leq \epsilon_2 ||P_m(A)|| = \epsilon_2 ||VP_m(\Lambda)V^{-1}||$$

$$||z_m|| \leq \epsilon_2 ||V|| ||P_m(\Lambda)|| ||V^{-1}||$$
$$\leq \epsilon_2 \kappa(V) \max_{\lambda \epsilon \rho(A)} |\lambda|$$

where $\kappa(V)$ is the condition number of matrix $V$, and its value is less than 1. Since matrix $V$ is a well conditioned. Since $\lambda$ is any eigenvalue of matrix $A$. Therefore equation (A.1.6) becomes

$$||x_{m+1} - x^*|| \leq \epsilon_1 + |A_{i+1}| \epsilon_2 \kappa(V) \max_{\lambda \epsilon \rho(A)} |\lambda| = \epsilon_1 + \epsilon_3 \kappa(V) \max_{\lambda \epsilon \rho(A)} |\lambda|$$

$$||x_{m+1} - x^*|| \leq \epsilon.$$

By following the same approach of section (4.1.1) for the convergence of Lanczos/Orthomin algorithm which is also called algorithm $A_8/B_{10}$ in C. Baheux [4].

## A.2 Tables for Monitoring Lanczos-type algorithm Chapter 4

### A.2.1 Monitoring Lanczos-type Algorithm based on relation $A_{12}$

**Table A.1:** *Behaviour of coefficients of $A_{12}$, on Baheux-type problems, when $\delta = 0.2$*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 | Col.6 | Col.7 | Col.8 |
|---|---|---|---|---|---|---|---|
| Dim. of $A$ | $k$ | $\Delta_{k+1}$ | $B_{k+1}$ | $C_{k+1}$ | $F_{k+1}$ | $G_{k+1}$ | $A_{k+1}$ |
| 100 | 143 | Inf | NaN | NaN | -4.0667E+01 | NaN | NaN |
| 500 | 138 | -Inf | NaN | NaN | -1.2363E+00 | NaN | NaN |
| 1000 | 138 | Inf | NaN | NaN | -1.0483E+02 | NaN | NaN |
| 5000 | 137 | Inf | NaN | NaN | 3.1365E+01 | NaN | NaN |
| 10000 | 136 | NaN | NaN | NaN | 9.5220E+01 | NaN | NaN |
| 15000 | 137 | -1.8339E+307 | Inf | NaN | -5.0930E+01 | -Inf | NaN |
| 20000 | 137 | NaN | NaN | NaN | -1.0087E+01 | NaN | NaN |
| 30000 | 138 | 1.4730E+308 | NaN | NaN | -3.2086E-01 | NaN | NaN |
| 40000 | 137 | NaN | NaN | NaN | 3.1354E+01 | NaN | NaN |
| 50000 | 137 | NaN | NaN | NaN | -8.4727E+01 | NaN | NaN |
| 60000 | 137 | -Inf | NaN | NaN | -6.8333E+00 | NaN | NaN |
| 70000 | 137 | 2.1116E+307 | NaN | NaN | -3.6980E+01 | NaN | NaN |
| 80000 | 137 | 7.0578E+307 | Inf | NaN | -5.9747E+00 | -Inf | NaN |
| 90000 | 98 | -6.1147E+203 | 2.8390E+01 | NaN | 0.0000E+00 | 5.2821E+01 | NaN |

**Table A.2:** *Behaviour of coefficients of $A_{12}$, on Baheux-type problems, when $\delta = 5$*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 | Col.6 | Col.7 | Col.8 |
|---|---|---|---|---|---|---|---|
| Dim. of $A$ | $k$ | $\Delta_{k+1}$ | $B_{k+1}$ | $C_{k+1}$ | $F_{k+1}$ | $G_{k+1}$ | $A_{k+1}$ |
| 100 | 110 | NaN | NaN | NaN | -1.3579E+01 | NaN | NaN |
| 500 | 110 | NaN | NaN | NaN | -1.8447E+00 | NaN | NaN |
| 1000 | 110 | NaN | NaN | NaN | 1.4353E+01 | NaN | NaN |
| 5000 | 110 | 1.0052E+307 | NaN | NaN | -4.7945E-01 | NaN | NaN |
| 10000 | 109 | NaN | NaN | NaN | -1.5253E+01 | NaN | NaN |
| 15000 | 108 | NaN | NaN | NaN | -3.6764E+00 | NaN | NaN |
| 20000 | 109 | 5.1738E+306 | -Inf | NaN | 1.3255E+01 | -Inf | NaN |
| 30000 | 111 | NaN | NaN | NaN | -3.6698E+00 | NaN | NaN |
| 40000 | 110 | NaN | NaN | NaN | -4.9091E+01 | NaN | NaN |
| 50000 | 108 | -7.9368E+306 | NaN | NaN | -3.1271E-02 | NaN | NaN |
| 60000 | 108 | 5.2744E+306 | NaN | NaN | 3.3261E-01 | NaN | NaN |
| 70000 | 107 | NaN | NaN | NaN | -1.0807E+01 | NaN | NaN |
| 80000 | 110 | -Inf | NaN | NaN | 8.1210E+01 | NaN | NaN |
| 90000 | 109 | 6.6387E+306 | NaN | NaN | 7.6687E+00 | NaN | NaN |

**Table A.3:** *Behaviour of coefficients of $A_{12}$, on Baheux-type problems, when $\delta = 8$*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 | Col.6 | Col.7 | Col.8 |
|---|---|---|---|---|---|---|---|
| Dim. of $A$ | $k$ | $\Delta_{k+1}$ | $B_{k+1}$ | $C_{k+1}$ | $F_{k+1}$ | $G_{k+1}$ | $A_{k+1}$ |
| 100 | 94 | -Inf | NaN | NaN | 9.4810E+00 | NaN | NaN |
| 500 | 95 | -1.2204E+308 | NaN | NaN | 1.7348E+00 | NaN | NaN |
| 1000 | 95 | NaN | NaN | NaN | -4.6667E+03 | NaN | NaN |
| 5000 | 94 | NaN | NaN | NaN | -1.7000E+01 | NaN | NaN |
| 10000 | 94 | NaN | NaN | NaN | 1.0163E+01 | NaN | NaN |
| 15000 | 94 | 2.3441E+307 | NaN | NaN | -2.5800E-01 | NaN | NaN |
| 20000 | 95 | NaN | NaN | NaN | 1.1804E+02 | NaN | NaN |
| 30000 | 94 | NaN | NaN | NaN | 5.5824E+00 | NaN | NaN |
| 40000 | 93 | NaN | NaN | NaN | -1.7267E+01 | NaN | NaN |
| 50000 | 93 | 1.6658E+306 | NaN | NaN | -4.6611E+01 | NaN | NaN |
| 60000 | 95 | NaN | NaN | NaN | -4.9448E+01 | NaN | NaN |
| 70000 | 94 | -4.4610E+306 | NaN | NaN | -6.0524E+01 | NaN | NaN |
| 80000 | 94 | 2.1993E+307 | -Inf | NaN | 2.9010E-01 | -Inf | NaN |
| 90000 | 94 | NaN | NaN | NaN | -5.5529E+01 | NaN | NaN |

**Table A.4:** *Behaviour of the parameters of the offending coefficients of $A_{12}$, on Baheux-type problems, when* $\delta = 0.2$

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 | Col.6 | Col.7 | Col.8 |
|---|---|---|---|---|---|---|---|
| Dim. of $A$ | $k$ | $a_{11}$ | $a_{13}$ | $a_{21}$ | $a_{23}$ | $a_{31}$ | $a_{33}$ |
| 100 | 143 | 1.2809E+103 | 3.1496E+101 | 1.4446E+104 | 4.4095E+102 | 1.5141E+105 | 5.1234E+103 |
| 500 | 138 | 2.7570E+102 | 2.2300E+102 | 3.6055E+103 | 3.4234E+103 | 6.6905E+104 | 4.6900E+104 |
| 1000 | 138 | 1.2209E+103 | 1.1647E+101 | 2.3236E+104 | -1.7673E+102 | 3.7150E+105 | -1.3911E+103 |
| 5000 | 137 | -6.6159E+102 | 2.1093E+101 | -8.6313E+103 | 4.6233E+102 | -1.1268E+105 | 6.6409E+103 |
| 10000 | 136 | 5.1234E+103 | -5.3806E+101 | 1.0482E+105 | -9.9214E+102 | 1.8966E+106 | -1.7246E+104 |
| 15000 | 137 | 1.9773E+102 | 3.8824E+100 | 4.2275E+103 | 6.2118E+101 | 7.6935E+104 | 5.9493E+102 |
| 20000 | 137 | 2.0298E+102 | 2.0123E+101 | 6.3133E+103 | 4.4095E+102 | 1.2375E+105 | 6.2713E+103 |
| 30000 | 138 | 5.2494E+101 | 1.6361E+102 | 9.6589E+102 | 4.3465E+103 | 4.9274E+103 | 9.3397E+104 |
| 40000 | 137 | -2.7157E+103 | 8.6615E+101 | -7.8951E+104 | 1.9528E+103 | -1.6314E+106 | 2.9845E+104 |
| 50000 | 137 | -4.0770E+102 | -4.8120E+100 | -1.3746E+104 | -4.4182E+101 | -2.7762E+105 | 7.5941E+102 |
| 60000 | 137 | -2.3316E+102 | -3.4121E+101 | -4.9834E+103 | -6.9292E+102 | -9.8381E+104 | -1.8828E+104 |
| 70000 | 137 | 1.9816E+102 | 5.3588E+100 | 5.8653E+103 | 1.0849E+102 | 9.9165E+104 | 1.7358E+103 |
| 80000 | 137 | 2.0648E+102 | 3.4559E+101 | 4.2835E+103 | 9.5889E+102 | 7.6039E+104 | 1.9402E+104 |
| 90000 | 98 | 0.0000E+00 | -6.0291E+66 | 3.1846E+68 | -1.8725E+68 | 8.4924E+68 | -3.5486E+69 |

**Table A.5:** *Behaviour of the parameters of the offending coefficients of $A_{12}$, on Baheux-type problems, when* $\delta = 5$

| *Col.1* | *Col.2* | *Col.3* | *Col.4* | *Col.5* | *Col.6* | *Col.7* | *Col.8* |
|---|---|---|---|---|---|---|---|
| *Dim. of $A$* | $k$ | $a_{11}$ | $a_{13}$ | $a_{21}$ | $a_{23}$ | $a_{31}$ | $a_{33}$ |
| 100 | 110 | -1.8058E+103 | -1.3298E+102 | -1.1423E+104 | 2.9537E+103 | 8.7798E+104 | 4.4795E+104 |
| 500 | 110 | -4.2549E+101 | -2.3065E+101 | -4.2090E+103 | -1.6387E+103 | -1.3707E+105 | -4.2877E+104 |
| 1000 | 110 | -2.6130E+102 | 1.8205E+101 | -1.1263E+104 | 6.3211E+102 | -9.2258E+104 | 4.5549E+103 |
| 5000 | 110 | 7.6554E+100 | 1.5967E+101 | -3.8758E+102 | 8.9371E+102 | -3.5219E+104 | 1.1899E+104 |
| 10000 | 109 | 2.2400E+102 | 1.4685E+101 | 8.0224E+103 | 5.6136E+102 | 6.8793E+104 | 6.8894E+103 |
| 15000 | 108 | -1.9601E+102 | -5.3314E+101 | -7.2652E+103 | -1.5652E+103 | -3.3624E+104 | -6.3133E+103 |
| 20000 | 109 | -2.6466E+101 | 1.9967E+100 | -2.1838E+103 | 9.4325E+100 | -4.7791E+104 | -1.8238E+103 |
| 30000 | 111 | 1.3239E+102 | 3.6076E+101 | 4.2144E+103 | 1.6207E+103 | 2.7656E+104 | 2.4866E+104 |
| 40000 | 110 | -1.2697E+103 | -2.5864E+101 | -3.1937E+104 | -1.2567E+103 | 2.2310E+105 | -1.0233E+104 |
| 50000 | 108 | -7.6622E+100 | -2.4503E+102 | -3.8456E+102 | -1.6995E+103 | -6.5245E+103 | 1.8808E+105 |
| 60000 | 108 | -1.3012E+101 | 3.9120E+101 | -7.6146E+102 | 1.1948E+103 | -1.1666E+104 | 2.1676E+103 |
| 70000 | 107 | -5.5172E+102 | -5.1050E+101 | -3.7042E+104 | -2.1840E+103 | -7.3643E+105 | -2.5000E+104 |
| 80000 | 110 | 2.9225E+102 | -3.5987E+100 | 6.4165E+103 | -1.5732E+102 | -1.0956E+105 | -1.5510E+103 |
| 90000 | 109 | 6.1251E+101 | -7.9871E+100 | 2.4999E+103 | -9.1752E+101 | 3.2900E+104 | 7.0396E+103 |

**Table A.6:** *Behaviour of the parameters of the offending coefficients of $A_{12}$, on Baheux-type problems, when $\delta = 8$*

| *Col.1* | *Col.2* | *Col.3* | *Col.4* | *Col.5* | *Col.6* | *Col.7* | *Col.8* |
|---|---|---|---|---|---|---|---|
| *Dim. of A* | $k$ | $a_{11}$ | $a_{13}$ | $a_{21}$ | $a_{23}$ | $a_{31}$ | $a_{33}$ |
| 100 | 94 | 1.1237E+102 | -1.1852E+101 | 4.2013E+103 | -1.6186E+102 | -1.5183E+104 | 6.3063E+103 |
| 500 | 95 | 4.2745E+100 | -2.4639E+100 | -6.7927E+103 | -1.8185E+102 | -2.8377E+105 | -4.7253E+103 |
| 1000 | 95 | 7.0922E+104 | 1.5198E+101 | 2.6040E+106 | 4.2108E+102 | -9.4050E+107 | -1.9920E+104 |
| 5000 | 94 | -1.9247E+102 | -1.1322E+101 | -1.1271E+104 | -3.7679E+102 | 3.1619E+103 | 1.3544E+104 |
| 10000 | 94 | -2.3044E+102 | 2.2675E+101 | -8.2123E+103 | 1.2106E+103 | 2.7346E+105 | -1.2306E+104 |
| 15000 | 94 | 7.5894E+100 | 2.9416E+101 | -5.2758E+102 | 1.2425E+103 | -5.0619E+104 | -1.7760E+104 |
| 20000 | 95 | -2.6689E+103 | 2.2610E+101 | -1.0952E+105 | 6.4725E+102 | 1.1355E+106 | -2.9162E+104 |
| 30000 | 94 | 1.6055E+102 | -2.8761E+101 | 2.8006E+103 | -1.1438E+103 | -3.7849E+105 | 2.2882E+104 |
| 40000 | 93 | -2.7785E+102 | -1.6091E+101 | -1.5526E+104 | -7.1012E+102 | 7.3789E+104 | 7.0175E+103 |
| 50000 | 93 | 1.1782E+102 | 2.5278E+100 | 6.7784E+103 | 1.7612E+102 | 2.3443E+104 | 2.3887E+103 |
| 60000 | 95 | 1.1910E+103 | 2.4086E+101 | 6.4289E+104 | 1.7077E+103 | -4.3989E+105 | 1.1068E+104 |
| 70000 | 94 | -8.7863E+101 | -1.4517E+100 | -2.9185E+103 | -8.6296E+101 | 9.1446E+104 | -3.3167E+102 |
| 80000 | 94 | -1.7167E+101 | 5.9176E+101 | -1.5722E+101 | 9.1937E+102 | 4.3609E+104 | -7.4905E+104 |
| 90000 | 94 | 3.4012E+102 | 6.1252E+100 | 6.4194E+103 | 3.7219E+102 | -5.7647E+105 | -9.6716E+103 |

## A.2.2 Monitoring Lanczos-type Algorithm based on relation $A_4$

**Table A.7:** *Behaviour of coefficients of $A_4$, on Baheux-type problems, when $\delta = 0.2$.*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $A_{k+1}$ | $B_{k+1}$ | $E_{k+1}$ |
| 100 | 353 | NaN | NaN | 1.1163E+01 |
| 500 | 348 | NaN | NaN | 1.3478E+01 |
| 1000 | 348 | NaN | NaN | -4.7831E-01 |
| 5000 | 348 | NaN | NaN | -1.3990E+01 |
| 10000 | 348 | NaN | NaN | -6.1613E+00 |
| 15000 | 348 | NaN | NaN | -2.1622E-01 |
| 20000 | 348 | NaN | NaN | 1.3158E+00 |
| 30000 | 348 | NaN | NaN | 1.3723E+00 |
| 40000 | 313 | 0.0000E+00 | -Inf | 0.0000E+00 |
| 50000 | 348 | NaN | NaN | -4.1474E+00 |
| 60000 | 337 | NaN | NaN | NaN |
| 70000 | 233 | 0.0000E+00 | -Inf | 0.0000E+00 |
| 80000 | 348 | NaN | NaN | -3.4657E-01 |
| 90000 | 348 | NaN | NaN | -1.2987E-01 |

**Table A.8:** *Behaviour of coefficients of $A_4$, on Baheux-type problems, when $\delta = 5$.*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $A_{k+1}$ | $B_{k+1}$ | $E_{k+1}$ |
| 100 | 298 | NaN | NaN | -2.2381E+00 |
| 500 | 297 | NaN | NaN | -1.8365E+00 |
| 1000 | 297 | NaN | NaN | -5.2904E+00 |
| 5000 | 297 | NaN | NaN | 1.6413E+00 |
| 10000 | 297 | NaN | NaN | -4.6932E+00 |
| 15000 | 295 | NaN | NaN | -4.9449E+00 |
| 20000 | 296 | NaN | NaN | 5.3936E+01 |
| 30000 | 296 | NaN | NaN | 1.0081E+01 |
| 40000 | 297 | NaN | NaN | NaN |
| 50000 | 297 | NaN | NaN | -3.0275E+00 |
| 60000 | 297 | NaN | NaN | -1.0351E+01 |
| 70000 | 292 | NaN | NaN | 2.0430E+02 |
| 80000 | 297 | NaN | NaN | -2.2461E+00 |
| 90000 | 297 | NaN | NaN | 6.4000E+01 |

**Table A.9:** *Behaviour of coefficients of $A_4$, on Baheux-type problems, when $\delta = 8$.*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $A_{k+1}$ | $B_{k+1}$ | $E_{k+1}$ |
| 100 | 256 | NaN | NaN | 8.0417E+00 |
| 500 | 256 | NaN | NaN | 2.9273E+01 |
| 1000 | 256 | NaN | NaN | 4.8840E+00 |
| 5000 | 256 | NaN | NaN | 7.1727E+00 |
| 10000 | 256 | NaN | NaN | 5.1344E-01 |
| 15000 | 256 | NaN | NaN | -1.1850E+00 |
| 20000 | 256 | NaN | NaN | -5.4779E+01 |
| 30000 | 256 | NaN | NaN | 1.0123E+02 |
| 40000 | 256 | NaN | NaN | 2.6048E+01 |
| 50000 | 255 | NaN | NaN | -3.0734E+01 |
| 60000 | 256 | NaN | NaN | 3.2115E+00 |
| 70000 | 254 | NaN | NaN | -1.9930E+02 |
| 80000 | 256 | NaN | NaN | 1.4115E+02 |
| 90000 | 256 | NaN | NaN | 3.4716E+00 |

**Table A.10:** *Behaviour of the parameters of the offending coefficients of $A_4$, on Baheux-type problems, when* $\delta = 0.2$

| *Col.1* | *Col.2* | *Col.3* | *Col.4* | *Col.5* | *Col.6* |
|---|---|---|---|---|---|
| *Dim. of A* | $k$ | $a_k$ | $b_k$ | $c_k$ | $d_k$ |
| 100 | 353 | 4.5681E+288 | -4.0923E+287 | NaN | -3.8068E+287 |
| 500 | 348 | -2.0998E+290 | 1.5579E+289 | NaN | -6.8065E+289 |
| 1000 | 348 | 2.7531E+290 | 5.7558E+290 | NaN | 7.4211E+291 |
| 5000 | 348 | -4.6238E+291 | -3.3050E+290 | NaN | -1.5672E+291 |
| 10000 | 348 | -5.8167E+289 | -9.4408E+288 | NaN | -1.6932E+290 |
| 15000 | 348 | 6.0908E+287 | 2.8170E+288 | NaN | 4.8727E+289 |
| 20000 | 348 | -9.7453E+290 | 7.4064E+290 | NaN | 6.5878E+291 |
| 30000 | 348 | 4.2940E+290 | -3.1292E+290 | NaN | -4.8617E+291 |
| 40000 | 313 | 0.0000E+00 | 2.2141E+261 | 1.3776E+262 | 2.2879E+262 |
| 50000 | 348 | 3.8397E+291 | 9.2580E+290 | NaN | 1.6128E+292 |
| 60000 | 337 | NaN | 4.5380E+279 | NaN | 3.6304E+280 |
| 70000 | 233 | 0.0000E+00 | 3.1953E+186 | 1.4794E+188 | -5.4389E+185 |
| 80000 | 348 | -2.9236E+290 | -8.4358E+290 | NaN | -7.6598E+291 |
| 90000 | 348 | -6.0908E+288 | -4.6899E+289 | NaN | -4.5316E+290 |

**Table A.11:** *Behaviour of the parameters of the offending coefficients of $A_4$, on Baheux-type problems, when* $\delta = 5$

| *Col.1* | *Col.2* | *Col.3* | *Col.4* | *Col.5* | *Col.6* |
|---|---|---|---|---|---|
| *Dim. of A* | $k$ | $a_k$ | $b_k$ | $c_k$ | $d_k$ |
| 100 | 298 | 9.1606E+290 | 4.0930E+290 | NaN | 6.4514E+291 |
| 500 | 297 | 4.5647E+292 | 2.4856E+292 | NaN | 2.2671E+293 |
| 1000 | 297 | -5.2231E+291 | -9.8728E+290 | NaN | -1.1289E+291 |
| 5000 | 297 | -1.1806E+292 | 7.1935E+291 | NaN | 2.1548E+293 |
| 10000 | 297 | 5.3596E+289 | 1.1420E+289 | NaN | 1.4984E+290 |
| 15000 | 295 | -9.2375E+292 | -1.8681E+292 | NaN | -2.3580E+292 |
| 20000 | 296 | 3.7422E+293 | -6.9383E+291 | NaN | 6.2455E+291 |
| 30000 | 296 | -8.4809E+292 | 8.4130E+291 | NaN | -7.4832E+291 |
| 40000 | 297 | NaN | 3.7111E+292 | NaN | 5.9874E+293 |
| 50000 | 297 | 1.1261E+289 | 3.7195E+288 | NaN | 9.2564E+289 |
| 60000 | 297 | 1.5247E+290 | 1.4729E+289 | NaN | 1.6116E+290 |
| 70000 | 292 | -6.9981E+292 | 3.4253E+290 | NaN | -8.9949E+292 |
| 80000 | 297 | -9.9810E+291 | -4.4437E+291 | NaN | -4.3663E+292 |
| 90000 | 297 | -2.9933E+293 | 4.6770E+291 | NaN | -1.6961E+293 |

**Table A.12:** *Behaviour of the parameters of the offending coefficients of $A_4$, on Baheux-type problems, when* $\delta = 8$

| *Col.1* | *Col.2* | *Col.3* | *Col.4* | *Col.5* | *Col.6* |
|---|---|---|---|---|---|
| *Dim. of A* | $k$ | $a_k$ | $b_k$ | $c_k$ | $d_k$ |
| 100 | 256 | -1.5047E+292 | 1.8711E+291 | NaN | 6.3345E+291 |
| 500 | 256 | -6.6790E+293 | 2.2816E+292 | NaN | 1.0601E+294 |
| 1000 | 256 | 1.3191E+293 | -2.7009E+292 | NaN | 1.0367E+293 |
| 5000 | 256 | 3.2712E+292 | -4.5607E+291 | NaN | -1.1118E+293 |
| 10000 | 256 | 1.0976E+294 | -2.1377E+294 | NaN | -3.6692E+295 |
| 15000 | 256 | 2.2040E+293 | 1.8600E+293 | NaN | 5.2107E+294 |
| 20000 | 256 | -6.1681E+293 | -1.1260E+292 | NaN | -2.4514E+293 |
| 30000 | 256 | 4.9756E+293 | -4.9152E+291 | NaN | -6.9766E+292 |
| 40000 | 256 | -6.9928E+294 | 2.6846E+293 | NaN | -7.6346E+293 |
| 50000 | 255 | -1.9062E+295 | -6.2023E+293 | NaN | -1.9187E+295 |
| 60000 | 256 | 1.0319E+293 | -3.2132E+292 | NaN | -5.0331E+293 |
| 70000 | 254 | 7.7573E+293 | 3.8923E+291 | NaN | -6.2404E+291 |
| 80000 | 256 | -7.2403E+295 | 5.1294E+293 | NaN | -1.3574E+295 |
| 90000 | 256 | 1.4881E+294 | -4.2866E+293 | NaN | -7.3728E+294 |

### A.2.3 Monitoring Lanczos-type Algorithm based on relation $A_8/B_6$

**Table A.13:** *Behaviour of coefficients of $A_8/B_6$, on Baheux-type problems, when* $\delta = 0.2$.

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $A_{k+1}$ | $C_{k+1}$ | $E_{k+1}$ |
| 100 | 131 | NaN | NaN | NaN |
| 500 | 171 | NaN | NaN | NaN |
| 1000 | 171 | NaN | NaN | NaN |
| 5000 | 170 | NaN | NaN | NaN |
| 10000 | 172 | NaN | NaN | NaN |
| 15000 | 167 | NaN | NaN | NaN |
| 20000 | 174 | NaN | NaN | NaN |
| 30000 | 172 | NaN | NaN | NaN |
| 40000 | 174 | NaN | NaN | NaN |
| 50000 | 173 | NaN | NaN | NaN |
| 60000 | 169 | NaN | NaN | NaN |
| 70000 | 175 | NaN | NaN | NaN |
| 80000 | 172 | NaN | NaN | NaN |
| 90000 | 170 | NaN | NaN | NaN |

**Table A.14:** *Behaviour of coefficients of $A_8/B_6$, on Baheux-type problems, when $\delta = 5$.*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $A_{k+1}$ | $C_{k+1}$ | $E_{k+1}$ |
| 100 | 152 | NaN | NaN | NaN |
| 500 | 153 | NaN | NaN | NaN |
| 1000 | 150 | NaN | NaN | NaN |
| 5000 | 152 | NaN | NaN | NaN |
| 10000 | 151 | NaN | NaN | NaN |
| 15000 | 152 | NaN | NaN | NaN |
| 20000 | 152 | NaN | NaN | NaN |
| 30000 | 152 | NaN | NaN | NaN |
| 40000 | 151 | NaN | NaN | NaN |
| 50000 | 151 | NaN | NaN | NaN |
| 60000 | 152 | NaN | NaN | NaN |
| 70000 | 152 | NaN | NaN | NaN |
| 80000 | 151 | NaN | NaN | NaN |
| 90000 | 152 | NaN | NaN | NaN |

**Table A.15:** *Behaviour of coefficients of $A_8/B_6$, on Baheux-type problems, when $\delta = 8$.*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $A_{k+1}$ | $C_{k+1}$ | $E_{k+1}$ |
| 100 | 134 | NaN | NaN | NaN |
| 500 | 132 | NaN | NaN | NaN |
| 1000 | 131 | NaN | NaN | NaN |
| 5000 | 131 | NaN | NaN | NaN |
| 10000 | 131 | NaN | NaN | NaN |
| 15000 | 131 | NaN | NaN | NaN |
| 20000 | 130 | NaN | NaN | NaN |
| 30000 | 131 | NaN | NaN | NaN |
| 40000 | 131 | NaN | NaN | NaN |
| 50000 | 131 | NaN | NaN | NaN |
| 60000 | 131 | NaN | NaN | NaN |
| 70000 | 132 | NaN | NaN | NaN |
| 80000 | 131 | NaN | NaN | NaN |
| 90000 | 131 | NaN | NaN | NaN |

**Table A.16:** *Behaviour of the parameters of the offending coefficients of $A_8/B_6$, on Baheux-type problems, when $\delta = 0.2$*

| *Col.1* | *Col.2* | *Col.3* | *Col.4* | *Col.5* | *Col.6* | *Col.7* |
|---|---|---|---|---|---|---|
| *Dim. of A* | $k$ | $a_k$ | $b_k$ | $c_k$ | $f_k$ | $e_k$ |
| 100 | 131 | 1.3670E+98 | NaN | -7.1434E+207 | -4.1336E+208 | NaN |
| 500 | 171 | -2.9678E+140 | NaN | NaN | NaN | NaN |
| 1000 | 171 | 8.8225E+140 | NaN | -1.8901E+292 | 7.5446E+292 | NaN |
| 5000 | 170 | 1.4692E+141 | NaN | -5.4262E+292 | -8.7330E+293 | NaN |
| 10000 | 172 | 5.4184E+140 | NaN | NaN | NaN | NaN |
| 15000 | 167 | 2.5975E+140 | NaN | 1.2349E+293 | -1.5281E+293 | NaN |
| 20000 | 174 | 2.1493E+143 | NaN | NaN | NaN | NaN |
| 30000 | 172 | -3.4297E+142 | NaN | -3.3524E+291 | -2.2901E+292 | NaN |
| 40000 | 174 | 1.6889E+144 | NaN | NaN | NaN | NaN |
| 50000 | 173 | -5.0909E+140 | NaN | -5.9875E+292 | -1.1227E+293 | NaN |
| 60000 | 169 | -5.4728E+139 | NaN | NaN | NaN | NaN |
| 70000 | 175 | -1.0496E+138 | NaN | -2.4948E+292 | 7.4766E+292 | NaN |
| 80000 | 172 | -1.5667E+140 | NaN | NaN | NaN | NaN |
| 90000 | 170 | 2.7628E+141 | NaN | NaN | NaN | NaN |

**Table A.17:** *Behaviour of the parameters of the offending coefficients of $A_8/B_6$, on Baheux-type problems, when $\delta = 5$*

| *Col.1* | *Col.2* | *Col.3* | *Col.4* | *Col.5* | *Col.6* | *Col.7* |
|---|---|---|---|---|---|---|
| *Dim. of A* | $k$ | $a_k$ | $b_k$ | $c_k$ | $f_k$ | $e_k$ |
| 100 | 152 | -4.9063E+141 | NaN | -3.6447E+291 | -5.2391E+292 | NaN |
| 500 | 153 | -2.5626E+147 | NaN | 3.0950E+296 | NaN | NaN |
| 1000 | 150 | 1.8401E+144 | NaN | -1.9097E+292 | -2.4271E+294 | NaN |
| 5000 | 152 | 1.2775E+144 | NaN | -5.6837E+293 | -1.5076E+295 | NaN |
| 10000 | 151 | 5.2808E+142 | NaN | 3.6867E+295 | NaN | NaN |
| 15000 | 152 | -1.6898E+145 | NaN | 1.2253E+294 | 4.1257E+295 | NaN |
| 20000 | 152 | -3.3016E+145 | NaN | -2.0971E+294 | -9.3010E+295 | NaN |
| 30000 | 152 | -1.0967E+143 | NaN | -2.3205E+294 | -6.2173E+295 | NaN |
| 40000 | 151 | -7.9640E+144 | NaN | 9.5420E+293 | 5.3354E+295 | NaN |
| 50000 | 151 | 7.0552E+144 | NaN | -5.0305E+293 | -2.4010E+295 | NaN |
| 60000 | 152 | 1.4548E+145 | NaN | -9.6633E+294 | NaN | NaN |
| 70000 | 152 | -1.5788E+143 | NaN | 1.3788E+296 | NaN | NaN |
| 80000 | 151 | 2.3020E+145 | NaN | 8.7519E+292 | 1.0234E+295 | NaN |
| 90000 | 152 | -6.9211E+144 | NaN | -1.3999E+294 | -4.8148E+295 | NaN |

**Table A.18:** *Behaviour of the parameters of the offending coefficients of $A_8/B_6$, on Baheux-type problems, when $\delta = 8$*

| *Col.1* | *Col.2* | *Col.3* | *Col.4* | *Col.5* | *Col.6* | *Col.7* |
|---|---|---|---|---|---|---|
| *Dim. of A* | *k* | $a_k$ | $b_k$ | $c_k$ | $f_k$ | $e_k$ |
| 100 | 134 | -9.0009E+143 | NaN | -2.4032E+292 | 2.6741E+293 | NaN |
| 500 | 132 | -2.2024E+146 | NaN | 4.3841E+295 | NaN | NaN |
| 1000 | 131 | -5.7515E+144 | NaN | 3.1145E+294 | NaN | NaN |
| 5000 | 131 | 1.6022E+147 | NaN | 4.6646E+292 | 6.2334E+295 | NaN |
| 10000 | 131 | -2.3100E+145 | NaN | 1.2057E+294 | 4.1242E+295 | NaN |
| 15000 | 131 | -6.0925E+145 | NaN | -3.4951E+294 | 2.4083E+294 | NaN |
| 20000 | 130 | -3.4270E+144 | NaN | 2.7443E+295 | 1.0791E+297 | NaN |
| 30000 | 131 | -3.7728E+146 | NaN | -7.5069E+293 | -5.8624E+295 | NaN |
| 40000 | 131 | 4.2808E+144 | NaN | -3.5174E+296 | -1.7240E+298 | NaN |
| 50000 | 131 | 3.5555E+146 | NaN | 2.6283E+294 | 3.2205E+296 | NaN |
| 60000 | 131 | -2.0559E+147 | NaN | -4.1755E+293 | 3.6314E+294 | NaN |
| 70000 | 132 | -1.3088E+147 | NaN | -9.0732E+296 | NaN | NaN |
| 80000 | 131 | -5.1805E+145 | NaN | -2.7581E+295 | -1.4887E+297 | NaN |
| 90000 | 131 | 9.6768E+145 | NaN | 4.8522E+295 | NaN | NaN |

### A.2.4 Monitoring Lanczos-type Algorithm based on relation $A_8/B_{10}$

**Table A.19:** *Behaviour of the parameters of the offending coefficients of $A_8/B_{10}$, on Baheux-type problems, when $\delta = 0.2$*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $A_k$ | $C_k$ | $B_k$ |
| 100 | 178 | NaN | NaN | NaN |
| 500 | 183 | NaN | NaN | NaN |
| 1000 | 178 | NaN | NaN | NaN |
| 5000 | 182 | NaN | NaN | NaN |
| 10000 | 176 | Inf | 0.0000E+00 | NaN |
| 15000 | 181 | NaN | NaN | NaN |
| 20000 | 181 | NaN | NaN | NaN |
| 30000 | 181 | NaN | NaN | NaN |
| 40000 | 184 | NaN | NaN | NaN |
| 50000 | 181 | NaN | NaN | NaN |
| 60000 | 120 | Inf | 0.0000E+00 | NaN |
| 70000 | 176 | NaN | NaN | NaN |
| 80000 | 172 | Inf | 0.0000E+00 | NaN |
| 90000 | 183 | NaN | NaN | NaN |

**Table A.20:** *Behaviour of the parameters of the offending coefficients of $A_8/B_{10}$, on Baheux-type problems, when $\delta = 5$*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $A_k$ | $C_k$ | $B_k$ |
| 100 | 158 | NaN | NaN | NaN |
| 500 | 155 | NaN | NaN | NaN |
| 1000 | 154 | NaN | NaN | NaN |
| 5000 | 153 | NaN | NaN | NaN |
| 10000 | 153 | NaN | NaN | NaN |
| 15000 | 153 | NaN | NaN | NaN |
| 20000 | 153 | NaN | NaN | NaN |
| 30000 | 153 | NaN | NaN | NaN |
| 40000 | 153 | NaN | NaN | NaN |
| 50000 | 152 | NaN | NaN | NaN |
| 60000 | 153 | NaN | NaN | NaN |
| 70000 | 153 | NaN | NaN | NaN |
| 80000 | 153 | NaN | NaN | NaN |
| 90000 | 153 | NaN | NaN | NaN |

**Table A.21:** *Behaviour of the parameters of the offending coefficients of $A_8/B_{10}$, on Baheux-type problems, when $\delta = 8$*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $A_k$ | $C_k$ | $B_k$ |
| 100 | 135 | NaN | NaN | NaN |
| 500 | 133 | NaN | NaN | NaN |
| 1000 | 133 | NaN | NaN | NaN |
| 5000 | 132 | NaN | NaN | NaN |
| 10000 | 132 | NaN | NaN | NaN |
| 15000 | 132 | NaN | NaN | NaN |
| 20000 | 132 | NaN | NaN | NaN |
| 30000 | 133 | NaN | NaN | NaN |
| 40000 | 132 | NaN | NaN | NaN |
| 50000 | 131 | NaN | NaN | NaN |
| 60000 | 132 | NaN | NaN | NaN |
| 70000 | 132 | NaN | NaN | NaN |
| 80000 | 132 | NaN | NaN | NaN |
| 90000 | 132 | NaN | NaN | NaN |

**Table A.22:** *Behaviour of the parameters of the offending coefficients of $A_8/B_{10}$, on Baheux-type problems, when $\delta = 0.2$*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $a_k$ | $b_k$ | $c_k$ |
| 100 | 178 | 1.5104E+138 | NaN | NaN |
| 500 | 183 | -2.3401E+145 | NaN | NaN |
| 1000 | 178 | 5.6782E+147 | NaN | NaN |
| 5000 | 182 | -6.8952E+146 | NaN | NaN |
| 10000 | 176 | -4.5252E+141 | 0.0000E+00 | NaN |
| 15000 | 181 | 2.7971E+148 | NaN | NaN |
| 20000 | 181 | -2.5773E+148 | NaN | NaN |
| 30000 | 181 | -7.2924E+147 | NaN | NaN |
| 40000 | 184 | 2.9969E+146 | NaN | NaN |
| 50000 | 181 | -2.4662E+147 | NaN | NaN |
| 60000 | 120 | -5.9679E+88 | 0.0000E+00 | NaN |
| 70000 | 176 | -2.6050E+138 | NaN | NaN |
| 80000 | 172 | -1.4973E+137 | 0.0000E+00 | NaN |
| 90000 | 183 | 6.2435E+144 | NaN | NaN |

**Table A.23:** *Behaviour of the parameters of the offending coefficients of $A_8/B_{10}$, on Baheux-type problems, when $\delta = 5$*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $a_k$ | $b_k$ | $c_k$ |
| 100 | 158 | -9.9506E+144 | NaN | NaN |
| 500 | 155 | 3.2130E+146 | NaN | NaN |
| 1000 | 154 | -3.3970E+147 | NaN | NaN |
| 5000 | 153 | 1.6221E+145 | NaN | NaN |
| 10000 | 153 | 4.1643E+145 | NaN | NaN |
| 15000 | 153 | -5.9692E+147 | NaN | NaN |
| 20000 | 153 | -4.9588E+144 | NaN | NaN |
| 30000 | 153 | 1.9475E+146 | NaN | NaN |
| 40000 | 153 | -2.0946E+147 | NaN | NaN |
| 50000 | 152 | 6.0330E+146 | NaN | NaN |
| 60000 | 153 | -2.3041E+148 | NaN | NaN |
| 70000 | 153 | 9.0311E+144 | NaN | NaN |
| 80000 | 153 | 1.4389E+147 | NaN | NaN |
| 90000 | 153 | 2.8325E+145 | NaN | NaN |

**Table A.24:** *Behaviour of the parameters of the offending coefficients of $A_8/B_{10}$, on Baheux-type problems, when $\delta = 8$*

| Col.1 | Col.2 | Col.3 | Col.4 | Col.5 |
|---|---|---|---|---|
| Dim. of $A$ | $k$ | $a_k$ | $b_k$ | $c_k$ |
| 100 | 135 | 2.3650E+148 | NaN | NaN |
| 500 | 133 | 2.4457E+146 | NaN | NaN |
| 1000 | 133 | -5.6127E+146 | NaN | NaN |
| 5000 | 132 | 7.3507E+146 | NaN | NaN |
| 10000 | 132 | 2.5989E+147 | NaN | NaN |
| 15000 | 132 | -1.6577E+147 | NaN | NaN |
| 20000 | 132 | 4.4906E+147 | NaN | NaN |
| 30000 | 133 | -9.2673E+146 | NaN | NaN |
| 40000 | 132 | -6.5140E+144 | NaN | NaN |
| 50000 | 131 | 1.0568E+145 | NaN | NaN |
| 60000 | 132 | 6.2999E+147 | NaN | NaN |
| 70000 | 132 | -3.1328E+147 | NaN | NaN |
| 80000 | 132 | -9.5886E+147 | NaN | NaN |
| 90000 | 132 | -6.8746E+146 | NaN | NaN |