الجمهورية الجزائرية الديمقراطية الشعبية
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي والبحث العلمي
MINISTÉRE DE l'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
جامعة عمار ثليجي بالأغواط
UNIVERSITÉ AMAR TÉLÉDJI DE LAGHOUAT
كـلـيـــــةالـعـلـــــوم
FACULTE DES SCIENCES
قـســـــم الـريـاضــيات
DÉPARTEMENT DE MATHÉMATIQUES

Mémoire de MASTER

Domaine : Mathématiques et Informatique
Filière : Mathématiques
Option : Analyse Mathématique

Présenté par :

Manal Kadraoui

THEME

Méthodes numériques pour la résolution de l'équation de sylvester

Soutenance publique devant le jury composé de :

Dr.Belacel Amar	Pr	Président
Dr.Abdesselam Nawel	M.C.B	Encadreur
Dr.Gagui Abdelmalek	M.C.B	Examinateur

Année Universitaire 2022/2023

Acknowledgements

I would first like to thank God the Almighty who gave me the strength and patience to accomplish this modest work.

I would like to express my gratitude to my teacher, Mrs. Abdesselam Nawel, for having supervised, guided, helped and advised me to achieve the elaboration of this dissertation.

I address my sincere thanks to the members of the jury, Mr. Belacel Amar, president of the jury, Mr. Gagui Abdelmalek, examiner, for agreeing to examine this work.

I thank all the teachers who taught me throughout my university course. I thank my very dear parents, who have always been there for me, and who have sacrificed themselves for my happiness and my success.

Finally, I thank my colleagues for their advice and assistance. To all these speakers, I offer my thanks, my respect and my gratitude.

Dedication

To my grandmother

To my dear parents

To my brother and my sisters

To all my family

To all those who are dear to me

I dedicate this memoir which is the fruit of my work

ملخص :

الهدف من هذا العمل ، دراسة معادلة سيلفستر ، التي لها دور مهم في انشاء مراقب ليونبرغر و في نظرية المراقبة،والتواصل . وتقديم نماذج التخفيض ، والطرق العددية من اجل حل معادلات تفاضلية.

الكلمات المفتاحية : **معادلة سيلفستر ، الطرق العددية ، مراقب ليونبرغر.**

Abstract

The objective of this work is to solve the Sylvester equation, which plays an important role in the construction of the observer of Luenberger and also in the theory of control, communication, reduction models and numerical methods for the resolution differential equations.

keywords :

Sylvester equation, Numerical methods, Observer of Luenberger.

Notations

Dans tout ce qui suit nous utilisons les notation suivantes :

$\mathbb{R}$	l'ensemble des nombres réels .
$\mathbb{C}$	l'ensemble des nombres complexes .
$M_{n\times m}(\mathbb{R})$	L'ensemble des matrices à valeur dans $\mathbb{R}$.
e_m	vecteur .
$\sigma(A)$	le spectre de A .
$\mathcal{D}(0,1)$	le cirle unitaire .
$\Re(\lambda)$	la partie réel de λ .
$\mathbb{P}$	l'ensemble des valeurs propres ,tq $\Re(\mu_i) < 0$.
$K_k(A,v)$	sous-espace de krylov d'ordre k associé à A .
GMRES	la généralisation de la méthode de minimisation du résidu .

Contents

Introduction

The objective of this work is to solve the Sylvester equation, which plays an important role in the construction of the Luenberger observer, and also in the theory of control communication, reduction models and numerical methods for the resolution differential equations [6].

We are interested in the role of the Sylvester equation in the construction of the Luenberguer observer associated with the following control system

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Dx(t) \end{cases}$$

with $A \in M_n(\mathbb{R})$, $B \in M_{n\times k}(\mathbb{R})$, $D \in M_{m\times n}(\mathbb{R})$, $y \in \mathbb{R}^m$, $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^k$, and who gives a approximation $z(t) \in \mathbb{R}^m$ of the state vector $x(t)$. He this is a variant of the classical Sylvester equation, and is of shape $AX - XH = CG$ Or $X \in M_{n\times m}(\mathbb{R})$, $H \in M_m(\mathbb{R})$ and $G \in M_m(\mathbb{R})$ are at

determine and matrices $A \in M_n(\mathbb{R})$, and $C \in M_{n\times m}(\mathbb{R})$ are given.

Among the methods proposed to solve the Sylvester equation, we find the Hessenberg-Schur method which consists in choosing. $H \in M_m(\mathbb{R})$ real Schur matrix and $G \in M_m(\mathbb{R})$ identity matrix I_m but the application of the latter is not easy; if A is wide and hollow [3]. It is therefore interesting to see other methods for this kind of situation (A wide and hollow) and it is for this reason that we apply the method of Saad and Datta [3].

Before proceeding to the resolution, note that the matrices H, G and X must satisfy the following conditions [6]:

1. H be stable

2. X well conditioned

3. the pair $\{H, G\}$ is controllable.

because if H is stable (all the elements of the spectrum of H are at negative real part) we can show that the approximation $z(t)$ of $x(t)$ obtained by the Luenberger observer satisfies $e(t) = z(t) - Xx(t)$ tends to 0 when t tends to ∞. In addition, the set of these three matrices contribute to a good approximation $z(t)$ of $x(t)$.

Note also that if the spectrum of A and the spectrum of H are disjoint, there exists a unique matrix X which satisfies the equation of Sylvester. Suppose that $G = I_m$ (without loss of generality, because the product of G and C is a matrix of type $n \times m$) and that the matrix C is of rank 1; the equation then becomes

$$AX - HX = ce_m^T,$$

which gives the idea of applying Arnoldi's process with a first vector v_1 such that $v_{m+1} = \alpha c$; we find this vector by exploiting the fact that all the vectors of the Arnoldi basis of the Krylov subspace associated with A and v_1 are such that $v_{i+1} = q_i(A)v_1$ where q_i is a polynomial of degree i.

The Arnoldi process then calculate, using v_1, the matrix H_m, upper Hessenberg matrix, which is not determined stable, we therefore proceed to a placement of poles for the stabilizer, which means that we present an algorithm in two main steps

1. find v_1 such that $v_{m+1} = \alpha c$ by finding the solution of $c = q_m(A)v_1$ by the GMRES method [12].

2. apply the pole placement algorithm, proposed by Saad and Datta [3] on the matrix H_m to stabilize it.

We can then calculate the solution X from the results obtained.

Chapter 1 be reserved for a few reminders, on the control theory, on the GMRES method and the Arnoldi process. In chapter 2 we come back to Sylvester's equation and we see how to exploit the Arnoldi process to solve it. In chapter 3 we explain how obtain v_1 such that $v_{m+1} = \alpha c$

In the fourth chapter, we present some numerical tests and for more details, the reader can consult the article of C.W Gear [5]. We end this work with a conclusion.

Chapter 1

Preliminary

In this chapter, we recall some notions on the theory of control, we then continue with reminders on the Arnoldi process [9] and we close this chapter with a reminder of the GMRES method [12].

Unless otherwise stated, throughout this chapter $A \in M_n(\mathbb{R})$,
$C \in M_{m\times n}(\mathbb{R})$, $B \in M_{n\times r}(\mathbb{R})$ and $x_0 \in \mathbb{R}^n$.

1.1 Control of linear systems

1.1.1 Dynamic Linear Systems

We will give the definition of a dynamic linear system.

Definition 1. *[10]:A discrete linear dynamical system admits the internal description as the following state representation:*

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k \end{cases} \tag{1.1.1}$$

where $x_k \in \mathbb{R}^n$ is the state vector and its dimension is that of the system;
$y_k \in \mathbb{R}^m$ output of system ; $u_k \in \mathbb{R}^r$ said system entry.

1.1.2 Continuous Dynamic Linear Systems

Definition 2. *[10]: A continuous linear dynamical system admits the internal description as following state representation:*

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \tag{1.1.2}$$

where $x(t) \in \mathbb{R}^n$ is the state vector and its dimension is that of the system; $y(t) \in \mathbb{R}^m$ system output; $u(t) \in \mathbb{R}^r$ said system entry.

1.2 Controllability and observability

In this section, we recall some very important notions in control theory.

1.2.1 Controllability

Definition 3. *[1] The system defined by (1.1.2) is said to be controllable if there exists a function $t \to u(t), 0 \leq t$, called control or command, allowing the system to pass from the initial state $x_0 = x(t_0)$ to any final state $x(t_1)$ $(0 \leq t \leq t_1 < \infty)$.*

Definition 4. *[1] The system defined by (1.1.1) is said to be controllable if there exists a finite family $\{u_0, u_1, ..., u_{N-1}\}$ allowing the passage of the system from the initial state x_0 to a final state x_N $(0 < N < \infty)$.*

Proposition 1. ***characterization of controllability*** *[10, 1]*

We consider the system defined by (1.1.2), the following properties are equivalent:

1. *the pair $\{A, B\}$ is said to be controllable*
2. *the matrix $\Delta = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}$ has maximum rank*
   i.e: $rg(\Delta) = n$ (criterion of Kalman's controllability)
3. *$rg(A - \lambda I, B) = n \ \forall \lambda \in \sigma(A)$ where $\sigma(A)$ is the spectrum of A.*

For discrete systems (defined by (1.1.1)) the controllability is given by the Kalman criterion [1].

1.2.2 Observability

Definition 5. *The system defined by (1.1.2) is observable, if for all* $0 \leq t \leq t_1$*, there exists* $t_1 > 0$ *such that the state initial* x_0 *can be uniquely determined from* $u(t)$ *and* $y(t)$*.*

Definition 6. *The system defined by (1.1.1) is observable, if there exists a rank* $N < \infty$ *such that the initial state* x_0 *can be entirely uniquely determined from the knowledge of the inputs* $\{u_0, u_1, ..., u_{N-1}\}$ *and the outputs* $y_0, ..., y_N$*.*

Proposition 2. ***Observability characterization*** *[10] A linear system (continuous or discrete) is said to be observable (or the pair* $\{A, C\}$ *is said to be observable) if the matrix defined by*

$\Gamma = \begin{bmatrix} C & CA & CA^2 & \dots & CA^{n-1} \end{bmatrix}$ *is of maximum rank i.e:*

$$\{A, C\} \text{ observable } \Leftrightarrow rg(\Gamma) = n$$

Proposition 3. *[10]**Duality***

The pair $\{A, C\}$ *is observable iff the pair* $\{A^T, C^T\}$ *is controllable.*

1.3 Stability

Discrete case: We consider the following linear system

$$\begin{cases} x_{k+1} = Ax_k \\ x_0 \text{ given} \end{cases} \tag{1.3.1}$$

Definition 7. *[1] We say that the system (1.3.2) is stable if the spectrum of the matrix* A *is inside the unit circle (in the complex plane) i.e:*

$$\sigma(A) \subset \mathcal{D}(0, 1)$$

Continuous case: We consider the following linear system

$$\begin{cases} \dot{x}(t) = Ax(t) \quad t > 0 \\ x(0) = x_0 \end{cases} \tag{1.3.2}$$

Definition 8. *[1] We say that a type system (1.3.2) is stable if all values of A have a negative real part. ie:*

$$(\forall \lambda \in \sigma(A)) \;\; \Re(\lambda) \in \mathbb{R}^{-}$$

1.4 Luenberger observer and Sylvester equation

In practice, we do not have access to all the components of the state $x(t)$ of the system (1.1.2) (nor to x_0) in a way explicit; we then construct an approximation $z(t)$ for the state vector $x(t)$. And it is in this horizon that we use the observer de Luenberger [1].

Definition 9. *[1] The Luenberger observer associated with the control system (1.1.2) is given by*

$$\begin{cases} \dot{z}(t) = Fz(t) + Dy(t) + Pu(t) \\ z(0) = z_0 \end{cases} \tag{1.4.1}$$

where $F, D \in M_m(\mathbb{R})$, $P \in M_{m\times k}(\mathbb{R})$ are to be determined.

the vector $z(t)$, given by (2.1.2), is an approximation of $x(t)$ if the error $e(t) = z(t) - Xx(t)$ tends to zero whatever the initial conditions $x(0)$, $z(0)$ and $u(t)$, which cannot take place, except if the conditions of the following theorem are verified.

Theorem 1. *[1] The system (1.4.1) is the observer system associated with the system (1.1.2), and $z(t)$ is an approximation of $Xx(t)$ in the sense that the error $e(t) = z(t) - Xx(t) \longrightarrow 0$ when $t \longrightarrow \infty$ for any initial condition $x(0)$, $z(0)$ and $u(t)$ if the following 3 conditions are verified:*

1) $XA - FX = DC$,

2) $P = XB$,

3) F *stable.*

proof. ,

By differentiating the relation $e(t) = z(t) - Xx(t)$ we obtain

$$\begin{aligned} \dot{e}(t) &= \dot{z}(t) - X\dot{x}(t) \\ &= Fz(t) + Dy(t) + Pu(t) - X(Ax(t) + Bu(t)), \end{aligned} \tag{1.4.2}$$

we use $y(t) = Cx(t)$ and we add and subtract $FXx(t)$ in (1.4.2), we get:

$$\dot{e}(t) = Fe(t) + (FX - XA + DC)x(t) + (P - XB)u(t), \tag{1.4.3}$$

if 1) and 2) hold the relation (1.4.3) becomes a first order differential equation given by

$$\dot{e}(t) = Fe(t)$$

and if 3) holds, we obviously have $\lim_{t\to\infty} e(t) = 0$ whatever the initial conditions $x(0)$,$z(0)$ and $u(t)$.□

And to determine the matrices F, D and P, we use Sylvester's equation; this is given by the following matrix equation

$$XA - FX = DC \tag{1.4.4}$$

where A and C are given, and X is the solution to be found. Its existence and uniqueness is ensured by the condition [6]:

$$\sigma(F) \cap \sigma(A) = \emptyset \tag{1.4.5}$$

We see in chapter 2 how to solve this equation by the Arnoldi process. □

1.5 The Partial Pole Placement Problem

We consider the control system (of type (1.1.2)) next

$$\begin{cases} \dot{x}(t) = Ax(t) + bu(t) \\ y(t) = Cx(t) \end{cases} \tag{1.5.1}$$

(where $b \in \mathbb{R}^n$) and we suppose that the vector $f \in \mathbb{R}^n$ is such that $u = -f^T x(t)$ thus the solution of the system (1.5.1) is written [7]

$$x(t) = exp[(A - bf^T)t]x_0 \tag{1.5.2}$$

it would therefore be desirable for $x(t)$ to be stable i.e.

$$\lim_{t \to +\infty} x(t) = 0 \tag{1.5.3}$$

we then consider $\sigma(A) = \{\lambda_i\}_{i=1}^n$ the spectrum of A and we suppose that

$$\Re(\lambda_i) \geq 0, \qquad \text{for } i = 1, ..., m \tag{1.5.4}$$

$$\Re(\lambda_i) < 0, \quad \text{for } i = m+1, ..., n \tag{1.5.5}$$

and let $\mathbb{P} = \{\mu_1, ..., \mu_m\} \subset \mathbb{C}$ with $\Re(\mu_i) < 0$. Thus (1.5.3) can only be verified if $A - bf^T$ is stable, in other words if we have (according to the definition 8)

$$\forall \lambda \in \sigma(A - bf^T) \;\; \Re(\lambda) \in \mathbb{R}^- \tag{1.5.6}$$

the choice of the vector $f \in \mathbb{R}^n$ must therefore be oriented towards a replacement of unstable eigenvalues i.e

$$\sigma(A - bf^T) = \mathbb{P} \cup \{\lambda_i\}_{i=m+1}^n \tag{1.5.7}$$

to verify the relation (1.5.3).

In summary we define the problem of partial placement of the poles as follows [7]

Definition 10. *Let* $\mathbb{P} = \{\mu_1, ..., \mu_m\} \subset \mathbb{C}$ *with* $(\forall \mu_i \in \mathbb{P})$ $\Re(\mu_i) < 0$ *and let* $b \in \mathbb{R}^n$ *we*

define the problem of partial placement of the poles of the matrix A by:

$$choose\ f \in \mathbb{R}^n\ such\ as\ \sigma(A - bf^T) = \mathbb{P} \cup \{\lambda_i\}_{i=m+1}^n$$

Remark 1. *When dealing with a discrete dynamical system, the pole placement problem consists in putting the eigenvalues around the origin (at inside the unit circle), i.e. we use the same definition 10 except that the set $\mathbb{P} \subset \mathcal{D}(0,1)$ and for the spectrum of A we have:*

$$\begin{aligned} |\lambda_i| &\geq 1, \qquad for\ i = 1, ..., m \\ |\lambda_i| &< 1, \quad for\ i = m+1, ..., n \end{aligned}$$

.

1.6 Arnoldi process

The Arnoldi process uses the Gram-Schmidt method [9] for the construction of the orthonormal basis of the subspace of Krylov. This process is often used in projection methods on Krylov subspaces in order to solve linear systems of big size. This section will be reserved for this process.

Definition 11. *[9] We call Krylov subspace of order k associated with A for $v \neq 0$ the subspace generated by the vectors $v, Av, ..., A^{k-1}v$ ie:*

$$K_k(A, v) = vect\{v, Av, ..., A^{k-1}v\}$$

Remark 2. *For $z \in K_k(A, v)$ we have*

$$z \in K_k(A, v) \Leftrightarrow \exists q \in \mathcal{P}_{k-1}\ mboxsuchthatz = q(A)v$$

An orthonormal basis of the Krylov subspace $K_k(A, v)$, is constructed by the Arnoldi process [9].

Algorithm 1. *Arnoldi's Process*

-choose $v \neq 0$ and calculate $v_1 = v/\|v\|$,

-for $j = 1, ..., k$ *do*

calculate $h_{i,j} = (Av_j, v_i)$ *for* $i = 1, ..., j$

calculate $\tilde{v}_j = Av_j - \sum_{i=1}^{j} h_{i,j} v_i$,

$h_{j+1,j} = \|\tilde{v}_j\|$,

if $h_{j+1,j} = 0$ *stop,*

$v_{j+1} = \frac{\tilde{v}_j}{h_{j+1,j}}$,

-End j.

We set

$V_k = [v_1, ..., v_k]$,

$H_k = (h_{i,j})_{(1 \leq i,j \leq k)}$ the upper Hessenberg matrix

and $\tilde{H}_k = \begin{pmatrix} H_k \\ h_{k+1,k} e_k^T \end{pmatrix}$ the upper Hessenberg matrix $(k+1) \times k$ (where $e_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$ is

the $k^{ième}$ vector of the canonical basis of $\mathbb{R}^k$).

We therefore have the following results [9].

$$AV_k = V_k H_k + h_{k+1,k} v_{k+1} e_k^T, \tag{1.6.1}$$

$$AV_k = V_{k+1} \tilde{H}_k, \tag{1.6.2}$$

$$V_k^T AV_k = H_k. \tag{1.6.3}$$

Proposition 4. *[4] the vectors* $\{v_1, ..., v_k\}$ *generated by the algorithm 1 constitute an orthonormal basis of the Krylov subspace* $K_k(A, v)$ *and we also have for* $i = 1, ..., k$:

$$v_i = q_{i-1}(A) v_1 \tag{1.6.4}$$

with q_{i-1} *a polynomial of degree* $i - 1$

In practice, we use the modified Arnoldi process because it is more stable [9].

Algorithm 2. *The Modified Arnoldi Process*

-choose any v, *calculate* $\beta = \|v\|$ *and* $v_1 = v/\beta$

-for $j = 1, ..., k$ *do*

$\tilde{v}_j = Av_j$

for $i = 1, ..., j$ *do*

$h_{i,j} = (\tilde{v}_j, v_i)$

$\tilde{v}_j = \tilde{v}_j - h_{i,j}v_i$

end i

$h_{j+1,j} = \|\tilde{v}_j\|$

if $h_{j+1,j} = 0$ *stop*

$v_{j+1} = v_j/h_{j+1,j}$

-end j.

1.6.1 minimal polynomial and the characteristic polynomial

Definition 12. *[4]* q *is the minimal polynomial of* A *for* v *if* q *is the polynomial of lesser degree verifying*

- *the leading coefficient of* q *is equal to 1*
- $q(A)v = 0$.

Remark 3. *-According to the Cayley-Hamilton theorem the minimal ploynomial of* A *for* v *is at most degree equal to* n.

Proposition 5. *[4] The Arnoldi process stops at step* m

($h_{m+1,m} = 0$*), where* m *is the degree of the minimal polynomial of* A *for* v

According to the proposition 4 the vectors v_i are written in the form $v_i = q_{i-1}(A)v_1$ where $q_{i-1}(A)$ is a polynomial of degree $i-1$. Let us denote by m the degree of the minimal polynomial from A for v_1 we have

$$q_{i-1}(t) = \frac{1}{h_{i,i-1}}(tq_{i-2}(t) - \sum_{j=1}^{i-1} h_{j,i-1}q_{j-1}(t)) \text{ pour tout } i = 1, ..., m, \tag{1.6.5}$$

and for the case where $i = m+1$ we have

$$q_m(t) = tq_{m-1}(t) - \sum_{i=1}^{m} h_{i,m}q_{i-1}(t).$$

There is a relation between the polynomial (1.6.5) and the characteristic polynomial of the upper Hessenberg matrix H_i of order i; this relation is given by the following theorem.

Theorem 2. *[4] Let m be the degree of the minimal polynomial of A for v, then for $i = 1, ..., m$ we have*

$$det(tI_i - H_i) = \beta_i q_i(t) \tag{1.6.6}$$

As a result, we have the following corollary

Corollary 1. *[4] for $i = 1, ..., m$ (m the degree of the minimal polynomial of A for v) the polynomial q_i is the polynomial characteristic of the intermediate matrix H_i (calculated by the algorithm 1) up to a multiplicative constant.*

This corollary gives the relation between the polynomial $q_i(t)$ such that

$v_{i+1} = q_i(A)v_1$ and the characteristic polynomial of the matrix H_i. This result allow us to start the method of Saad and Datta [3] for solving the equation of Sylvester.

1.7 GMRES method and its implementation

We consider the linear system

$$Ax = b, \tag{1.7.1}$$

where $b \in \mathbb{R}^n$ and the matrix A is assumed to be real and reversible.

1.7.1 GMRES method [12]

The GMRES method (Generalized Minimal Residual [12]) is designed to solve the system (1.7.1). She is defined by:

given x_0 any approximation of x, solution of the system (1.7.1), and $r_0 = b - Ax_0$ the residual associated with x_0. At step $k \leq m$ (m being the degree of the minimal polynomial of A for $v_1 = r_0/\|r_0\|$), find the vector x_k, approximation of x, such that

$$x_k \in x_0 + K_k(A, r_0), \tag{1.7.2}$$

$$r_k \perp AK_k(A, r_0). \tag{1.7.3}$$

We use the Arnoldi process to build a base orthonormal of the Krylov subspace $K_k(A, r_0)$, v_1,...,v_k ($r_k = b - Ax_k$ is the residual associated with x_k) and we set:

$$V_k = [v_1, ..., v_k],$$

so (1.7.2) becomes

$$x_k = x_0 + V_k\alpha,$$

with $\alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{pmatrix} \in \mathbb{R}^k$, and the relation (1.7.3) translates to

$$\begin{aligned} & (AV_k)^T r_k = 0 \\ \Leftrightarrow \quad & (AV_k)^T (b - Ax_k) = 0 \\ \Leftrightarrow \quad & (AV_k)^T (b - A(x_0 + V_k\alpha)) = 0 \\ \Leftrightarrow \quad & (AV_k)^T r_0 - (AV_k)^T AV_k\alpha = 0 \\ \Leftrightarrow \quad & (AV_k)^T AV_k\alpha = (AV_k)^T r_0, \end{aligned}$$

or $(AV_k)^T AV_k$ is always invertible so

$$\alpha = [(AV_k)^T AV_k]^{-1}(AV_k)^T r_0,$$

Which give

$$x_k = x_0 + V_k[(AV_k)^T AV_k]^{-1}(AV_k)^T r_0.$$

Proposition 6. *[12] For the GMERS method, we have:*

$$\begin{aligned} r_k \perp AK_k(A, r_0) \Leftrightarrow \alpha \quad &= \quad argmin_{y \in \mathbb{R}^k} \|r_0 - AV_k y\| \\ &= \quad [(AV_k)^T AV_k]^{-1}(AV_k)^T r_0, \end{aligned}$$

Remark 4. *The proposition 6 shows that the orthogonality problem is leads to a minimiza-*

tion problem.

Algorithm 3. *Algorithm of the GMRES method*

1. *choose* x_0 *then calculate* $r_0 = b - Ax_0$, $\beta = \|r_0\|$ *and* $v_1 = r_0/\beta$
2. *use modified Arnoldi process to calculate* $v_1, ..., v_k$ *and* $\tilde{H}_k$
3. *solve* $argmin_{y \in \mathbb{R}^k} \|\beta e_1 - \tilde{H}_k y\| = y_k$
4. *calculate* $x_k = x_0 + V_k y_k$

1.7.2 Implementation of the GMRES method

To solve the minimization problem (step 3 Algorithm 3) we has recourse (according to the work of Saad and Schultz [12]), to Givens rotations to decompose the matrix $\tilde{H}_k$ as product of a triangular matrix $\tilde{R}_k$ and an orthogonal matrix Q_k.

If we carry out such a decomposition we have

$$\tilde{H}_k = Q_k \tilde{R}_k,$$

with $Q_k \in M_{k+1 \times k+1}(\mathbb{R})$, $\tilde{R}_k = \begin{pmatrix} R_k \\ 0^T \end{pmatrix}$ and R_k triangular of order k.

So the minimization problem of step 3 of the algorithm 3 gives

$$\begin{aligned} \beta e_1 - \tilde{H}_k y &= \beta e_1 - Q_k \tilde{R}_k y, \\ &= \beta Q_k Q_k^T e_1 - Q_k \tilde{R}_k y \qquad (\text{car } Q_k Q_k^T = I_{k+1}), \\ &= Q_k(\beta Q_k^T e_1 - \tilde{R}_k y), \end{aligned}$$

Q_k is orthogonal so keeps the norm, then we have

$$\begin{aligned} \|\beta e_1 - \tilde{H}_k y\| &= \|Q_k(\beta Q_k^T e_1 - \tilde{R}_k y)\| \\ &= \|(\beta Q_k^T e_1 - \tilde{R}_k y)\|, \end{aligned}$$

we set $Q_k^T(\beta e_1) = \tilde{g}_k = \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_{k+1} \end{pmatrix}$ and $g_k = \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_k \end{pmatrix}$

we obtain

$$\begin{aligned} \|\tilde{g}_k - \tilde{R}_k y\|^2 &= \| \begin{pmatrix} g_k \\ \gamma_{k+1} \end{pmatrix} - \begin{pmatrix} R_k y \\ 0 \end{pmatrix} \|^2 \\ &= \| \begin{pmatrix} g_k - R_k y \\ \gamma_{k+1} \end{pmatrix} \|^2 \\ &= \|g_k - R_k y\|^2 + |\gamma_{k+1}|^2, \end{aligned}$$

by minimizing we get

$$y_k = R_k^{-1} g_k \in \mathbb{R}^k. \tag{1.7.4}$$

To decompose the matrix $\tilde{H}_k$ we define the Givens rotation matrix Ω_i

$$\Omega_i = \begin{pmatrix} 1 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \ddots & \ddots & \dots & \dots & \dots & \dots & \vdots \\ \vdots & \ddots & 1 & 0 & 0 & 0 & \dots & \vdots \\ \vdots & \dots & 0 & c_i & s_i & 0 & \dots & \vdots \\ \vdots & \dots & 0 & -s_i & c_i & 0 & \dots & \vdots \\ \vdots & \dots & 0 & 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \dots & \dots & \dots & \dots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & 1 \end{pmatrix},$$

where the s_i and c_i are such that $s_i^2 + c_i^2 = 1$ ($c_i = cos\theta_i$, $s_i = sin\theta_i$).

We set $\tilde{H}_k^{(0)} = \tilde{H}_k$ and for $i = 1, ..., k$ we have

$$\Omega_i ... \Omega_2 \Omega_1 \tilde{H}_k^{(0)} = \tilde{H}_k^{(i)} = (h_{l,m}^{(i)}),$$

and we put

$$Q_k^T = \Omega_k \Omega_{k-1} ... \Omega_1,$$

thus the resolution of the minimization problem is done in stages as follows

First Step

we calculate

$$\begin{cases} c_1 = \frac{h_{1,1}}{\sqrt{(h_{1,1}^2+h_{2,1}^2)}} \\ s_1 = \frac{h_{2,1}}{\sqrt{(h_{1,1}^2+h_{2,1}^2)}} \end{cases}$$

so

$$\begin{aligned} Q_1^T(\beta e_1) &= \begin{pmatrix} c_1 & s_1 & 0 & \dots & 0 \\ -s_1 & c_1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & 1 \end{pmatrix} \begin{pmatrix} \beta \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} c_1\beta \\ s_1\beta \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} \gamma_1^{(1)} \\ \gamma_2^{(1)} \\ \vdots \\ 0 \end{pmatrix} \\ &= g_1, \end{aligned}$$

$$
\begin{aligned}
Q_1^T(\widetilde{H}_k^{(0)}) &= \Omega_1 \widetilde{H}_k^{(0)} \\
&= \widetilde{H}_k^{(1)} \\
&= \begin{pmatrix}
h_{11}^{(1)} & h_{12}^{(1)} & \dots & \dots & h_{1k}^{(1)} \\
0 & h_{22}^{(1)} & \dots & \dots & h_{2k}^{(1)} \\
\vdots & h_{32}^{(1)} & \ddots & \dots & h_{3k}^{(1)} \\
\vdots & 0 & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & h_{kk}^{(1)} \\
0 & 0 & \dots & 0 & h_{k+1,k}^{(1)}
\end{pmatrix}
\end{aligned}
$$

Second Step

in the same way we calculate

$$\begin{cases} c_2 = \frac{h_{2,2}^{(1)}}{\sqrt{((h_{2,2}^{(1)})^2+(h_{3,2}^{(1)})^2)}} \\ s_2 = \frac{h_{3,2}^{(1)}}{\sqrt{((h_{2,2}^{(1)})^2+(h_{3,2}^{(1)})^2)}} \end{cases}$$

then

$$\begin{aligned} Q_2^T(\beta e_1) &= \Omega_2(\Omega_1(\beta e_1)) \\ &= \Omega_2(Q_1^T(\beta e_1)) \\ &= \Omega_2 g_1 \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & c_2 & s_2 & 0 & \dots & 0 \\ 0 & -s_2 & c_2 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & 1 \end{pmatrix} \begin{pmatrix} \gamma_1^{(1)} \\ \gamma_2^{(1)} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} \gamma_1^{(1)} \\ c_2\gamma_2^{(1)} \\ -s_2\gamma_2^{(1)} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} \gamma_1^{(2)} \\ \gamma_2^{(2)} \\ \gamma_3^{(2)} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\ &= g_2, \end{aligned}$$

so

$$\begin{cases} \gamma_1^{(2)} = \gamma_1^{(1)}, \\ \gamma_2^{(2)} = c_2\gamma_2^{(1)}, \\ \gamma_3^{(2)} = -s_2\gamma_2^{(1)}, \end{cases}$$

$$\begin{aligned} Q_2^T(\widetilde{H}_k^{(0)}) &= \Omega_2\widetilde{H}_k^{(1)} \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & c_2 & s_2 & 0 & \dots & 0 \\ 0 & -s_2 & c_2 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & 1 \end{pmatrix} \begin{pmatrix} h_{11}^{(1)} & h_{12}^{(1)} & \dots & \dots & h_{1k}^{(1)} \\ 0 & h_{22}^{(1)} & \dots & \dots & h_{2k}^{(1)} \\ \vdots & h_{32}^{(1)} & \ddots & \dots & h_{3k}^{(1)} \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & h_{kk}^{(1)} \\ 0 & 0 & \dots & 0 & h_{k+1,k}^{(1)} \end{pmatrix} \\ &= \begin{pmatrix} h_{11}^{(2)} & h_{12}^{(2)} & h_{13}^{(2)} & \dots & \dots & h_{1k}^{(2)} \\ 0 & h_{22}^{(2)} & h_{23}^{(2)} & \dots & \dots & h_{2k}^{(2)} \\ \vdots & 0 & h_{33}^{(2)} & \dots & \dots & h_{3k}^{(2)} \\ \vdots & \vdots & h_{43}^{(2)} & \ddots & \dots & \vdots \\ \vdots & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & h_{kk}^{(2)} \\ 0 & 0 & 0 & \dots & 0 & h_{k+1,k}^{(2)} \end{pmatrix} \\ &= \widetilde{H}_k^{(2)} \end{aligned}$$

in general at any step i we have

$$\begin{aligned} Q_i^T(\beta e_1) &= \Omega_i(\Omega_{i-1}...\Omega_1(\beta e_1)) \\ &= \Omega_i(Q_{i-1}^T(\beta e_1)) \\ &= \Omega_i g_{i-1} \end{aligned}$$

$$
\begin{aligned}
&= \begin{pmatrix}
1 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\
0 & \ddots & \ddots & \dots & \dots & \dots & \dots & \vdots \\
\vdots & \ddots & 1 & 0 & 0 & 0 & \dots & \vdots \\
\vdots & \dots & 0 & c_i & s_i & 0 & \dots & \vdots \\
\vdots & \dots & 0 & -s_i & c_i & 0 & \dots & \vdots \\
\vdots & \dots & 0 & 0 & 0 & 1 & \ddots & \vdots \\
\vdots & \dots & \dots & \dots & \dots & \ddots & \ddots & 0 \\
0 & \dots & \dots & \dots & \dots & \dots & 0 & 1
\end{pmatrix}
\begin{pmatrix}
\gamma_1^{(i-1)} \\ \gamma_2^{(i-1)} \\ \vdots \\ \gamma_{i-1}^{(i-1)} \\ \gamma_i^{(i-1)} \\ 0 \\ \vdots \\ 0
\end{pmatrix} \\
&= \begin{pmatrix}
\gamma_1^{(i)} \\ \gamma_2^{(i)} \\ \vdots \\ c_i\gamma_i^{(i-1)} \\ -s_i\gamma_i^{(i-1)} \\ 0 \\ \vdots \\ 0
\end{pmatrix} \\
&= g_i,
\end{aligned}
$$

with

$$
\begin{cases}
s_i = \frac{h_{i+1,i}^{(i-1)}}{\sqrt{(h_{i,i}^{(i-1)})^2+(h_{i+1,i}^{(i-1)})^2}}, \\
c_i = \frac{h_{i,i}^{(i-1)}}{\sqrt{(h_{i,i}^{(i-1)})^2+(h_{i+1,i}^{(i-1)})^2}},
\end{cases}
$$

where the s_i and c_i are such that $s_i^2 + c_i^2 = 1$ ($c_i = cos\theta_i$, $s_i = sin\theta_i$).

the vector g_i is given by

$$g_i = \begin{pmatrix} \gamma_1^{(i)} \\ \gamma_2^{(i)} \\ \vdots \\ \gamma_i^{(i)} \\ \gamma_{i+1}^{(i)} \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

with

$$\begin{cases} \gamma_j^{(i)} = \gamma_j^{(i-1)} & j = 1, ..., i-1, \\ \gamma_i^{(i)} = c_i \gamma_i^{(i-1)}, \\ \gamma_{i+1}^{(i)} = -s_i \gamma_i^{(i-1)}, \end{cases}$$

only the components $\gamma_i^{(i)}$ and $\gamma_{i+1}^{(i)}$ which change at every step i.

$$\begin{aligned} Q_i^T(\widetilde{H}_k^{(i-1)}) &= \Omega_i \widetilde{H}_k^{(i-1)} \\ &= \widetilde{H}_k^{(i)} \\ &= \begin{pmatrix} h_{11}^{(i)} & \dots & h_{1i}^{(i)} & h_{1,i+1}^{(i)} & \dots & h_{1k}^{(i)} \\ 0 & \ddots & \vdots & \vdots & \dots & \vdots \\ \vdots & \ddots & h_{ii}^{(i)} & \vdots & \dots & \vdots \\ \vdots & \ddots & 0 & h_{i+1,i+1}^{(i)} & \dots & \vdots \\ \vdots & \ddots & 0 & h_{i+2,i+1}^{(i)} & \ddots & \vdots \\ \vdots & \ddots & \vdots & \ddots & \ddots & h_{kk}^{(i)} \\ 0 & \dots & 0 & \dots & 0 & h_{k+1,1}^{(i)} \end{pmatrix} \end{aligned}$$

at any step i only the terms $h_{i,i}^{(i)}$ and $h_{i+1,i}^{(i)}$ will be modified:

$$\begin{cases} h_{i,i}^{(i)} = c_i h_{i,i}^{(i-1)} + s_i h_{i,i+1}^{(i-1))}, \\ h_{i+1,i}^{(i)} = 0. \end{cases}$$

Thus the vector y_k, solution of the minimization problem (at step 3 of the algorithm 3), is obtained from the following equation

$$R_k y_k = g_k,$$

which confirms the relation given by (1.7.4).

Conclusion

We have seen in this chapter reminders for the resolution of the Sylvester equation given by $AX - XH = CG$, in particular the condition of uniqueness and existence of the solution X. We introduce in the next chapter the problem posed by this equation and we return in detail to its resolution using the Arnoldi process.

Chapter 2

Resolution Sylvester's Equation

In this chapter, we will present the work of Saad and Datta [3]. The goal is to solve Sylvester's equation

$$\begin{cases} AX - XH = CG, \\ \sigma(H) = \{\mu_1, ..., \mu_m\}, \end{cases} \tag{2.0.1}$$

by applying the Arnoldi process, where $A \in M_n(\mathbb{R})$, $C \in M_{n\times m}(\mathbb{R})$ are given and $H \in M_m(\mathbb{R})$, $X \in M_{n\times m}(\mathbb{R})$, $G \in M_m(\mathbb{R})$ are to be found.

We will also see how to obtain the second equality given by the above expression.

Unless otherwise stated, in what follows the matrices A, X, H, C, G are defined as above and $\mathbb{P} = \{\mu_1, ..., \mu_m\}$ is an arbitrary set of m complex numbers with negative real part and different in pairs, where m is sufficiently less than n.

2.1 Position of the problem

We saw in chapter 1, that the construction of the Luenberger observer requires the resolution of the Sylvester equation. For to solve this equation one can use the Hessenberg-Schur method [3], but the latter does not give good results if the matrix A is large, hence the use of Arnoldi's method.

We consider the type system (1.1.2)

$$\begin{cases} \dot{x}(t) = Mx(t) + Bu(t), \\ y(t) = Ex(t), \end{cases} \tag{2.1.1}$$

Or $E \in M_{m\times n}(\mathbb{R})$, $M \in M_n(\mathbb{R})$, $B \in M_{n\times k}(\mathbb{R})$, $x(t) \in \mathbb{R}^n$ $u(t) \in \mathbb{R}^k$ and $y(t) \in \mathbb{R}^m$.

The Luenberger observer [1] associated with this system is given by

$$\begin{cases} \dot{z}(t) = Fz(t) + Dy(t) + Pu(t), \\ z(0) = z_0, \end{cases} \tag{2.1.2}$$

where the matrices $F, D \in M_m(\mathbb{R})$, $P \in M_{m\times k}(\mathbb{R})$ are to be determined.

According to the theorem 1, seen in chapter 1, the error between $z(t)$ and $x(t)$ given by

$$e(t) = z(t) - Yx(t), \tag{2.1.3}$$

with $Y \in M_{m,n}(\mathbb{R})$, tends to 0 at infinity if the three following conditions are verified

1. $YM - FY = DE$,
2. $P = YB$,
3. F stable.

Condition 1 is none other than Sylvester's equation, and as we saw in chapter 1, it admits a unique solution if

$$\sigma(F) \cap \sigma(M) = \emptyset, \tag{2.1.4}$$

to find the relation (2.0.1) we put

$$M = A^T,\ F = H^T,\ E = C^T,\ D = G^T,\ Y = X^T$$

thus condition 1 becomes

$$X^T A^T - H^T X^T = G^T C^T, \tag{2.1.5}$$

that's to say

$$AX - XH = CG, \tag{2.1.6}$$

therefore the condition of existence and uniqueness of the solution given by (1.4.5) becomes

$$\sigma(H) \cap \sigma(A) = \emptyset. \tag{2.1.7}$$

As regards the matrix G, we choose it so that the pair $\{H, G\}$ is controllable [7].

Thus the conditions required to solve Sylvester's equation are

1. H stable,
2. $\sigma(H) \cap \sigma(A) = \emptyset$,
3. the controllable $\{H, G\}$ pair.

Since the product of the matrices G and C gives a matrix of type $n \times m$ we can assume in all that follows, and without loss of generality, that the matrix G is the identity matrix; so the equation to solve is

$$\begin{cases} AX - XH = C, \\ \sigma(H) = \mathbb{P}. \end{cases} \tag{2.1.8}$$

Note that if we multiply C by a scalar then the new solution is obtained by multiplying the matrix X by the same scalar.

In what follows we will study the problem in the case where C is of rank 1.

2.2 Application of the Arnoldi process

The application of the Arnoldi process starts from the analogy presented by the latter with the Sylvester equation given by (2.1.8), and this analogy is the basis of the method of Saad and Datta [3]. And this is what we do in what follows.

2.2.1 Saad and Datta method[3]

Since C has rank 1, we can write it in the form

$$C = (0, 0, ..., c), \tag{2.2.1}$$

so we can render it as a matrix product

$$C = ce_m^T, \tag{2.2.2}$$

the Sylvester equation is therefore written in the form

$$AX - XH = ce_m^T, \tag{2.2.3}$$

which reminds us of the Arnoldi process. In effect, this last is written (at step m) as follows:

$$AV_m - V_m H_m = h_{m+1,m} v_{m+1} e_m^T, \tag{2.2.4}$$

so we deduce that to solve Sylvester's equation, it is enough to find a vector v_1 such that the second member is ce_m^T after running m steps of Arnoldi's process (m less than the degree of the minimal polynomial of A for v_1).

On the other hand the resolution of the equation (2.1.8) also requires that $\sigma(H) = \mathbb{P}$, that is to say that the polynomial characteristic of the matrix H_m must be

$$q_m(t) = (t - \mu_1)...(t - \mu_m), \tag{2.2.5}$$

Since we have

$$p_m(A)v_1 = v_{m+1},$$

with $p_m(t) = \alpha det(tI_m - H_m)$ (according to the corollary 1, page 19), and knowing that the characteristic polynomial of the matrix H_m is written

$$q_m(t) = det(tI_m - H_m),$$

we deduce that

$$v_{m+1} = \alpha q_m(A)v_1,$$

Since we seek to have $v_{m+1} = \beta c$ with $\beta \in \mathbb{R}$, we deduce that v_1 must be proportional to c, indeed

$$\begin{aligned} \alpha q_m(A)v_1 &= \beta c \\ v_1 &= \underbrace{\frac{\beta}{\alpha}}\ [q_m(A)]^{-1}c \\ v_1 &= \gamma\ \ [q_m(A)]^{-1}c \end{aligned} \tag{2.2.6}$$

where γ is the normalization constant.

So the basic idea is to run Arnoldi's process starting with v_1 (the vector obtained by the relation (2.2.6)) then do a placement of poles on the matrix H_m, to have the desired spectrum, and that consists in finding $y \in \mathbb{R}^m$ such that

$$\sigma(H) = \sigma(H_m - ye_m^T) = \mathbb{P}, \tag{2.2.7}$$

Thus solving Sylvester's equation consists in solving two problems which are

1. find v_1 solution of $q_m(A)x = c$
2. find $y \in \mathbb{R}^m$ such that the relation (2.2.7) holds.

The first step consists in solving $q_m(A)x = c$ i.e. the resolution of m systems namely $(A - \mu_i I)x_i = x_{i-1}$ which is not easy. We give more details, in chapter 3, about the resolution of such systems. The choice of the set $\mathbb{P}$ must be done in such a way that its elements are not close to the eigenvalues of the matrix A, because in this case the solution x obtained be badly conditioned [6]. We deal in Chapter 3 with the resolution of this problem.

The placement of the poles of the matrix H_m is done with a simple procedure that we see in the next section.

2.3 Principle of the method of Saad and Datta

We present here the basic ideas of the two main stages of the method of Saad and Datta which are the assignment of poles of the matrix H_m and the choice of the first vector v_1

Definition 13. *Let $E = \{\alpha_1, ..., \alpha_j\}$. We say that the set E is stable by conjugation if:*

$$(\forall \alpha_j \in E) :\ \bar{\alpha_j} \in E$$

Theorem 3. *[6] Let H_m be the upper Hessenberg matrix obtained by the Arnoldi process. We denote by $\sigma(H_m)$ the spectrum of the matrix H_m. We assume that $\mathbb{P} \cap \sigma(H_m) = \emptyset$,*

$$s = \prod_{j=1}^{m} (H_m - \mu_j I)e_1, \quad \alpha = \prod_{j=1}^{m-1} h_{j+1,j}^{-1} \tag{2.3.1}$$

then $\sigma(H_m - \alpha s e_m^T) = \mathbb{P}$ and the matrix $H_m - \alpha s e_m^T$ is Superior Hessenberg.

Moreover if $\mathbb{P}$ is stable by conjugation then $\sigma(H_m - \alpha s e_m^T)$ is real.

We use the expression given by (2.2.4), we have the following lemma

Lemma 2.3.1. *[6] We consider the matrices V_m, H_m given by the Arnoldi process and we assume that $c \in K_{m+1}(A, v_1)$ and $v_{m+1}^T c \neq 0$ then $\exists f \in \mathbb{R}^m$ and $\beta_m \in \mathbb{R}$ such as:*

$$AV_m - V_m(H_m - f e_m^T) = \beta_m c e_m^T \tag{2.3.2}$$

proof. ,
Let $\beta_m = h_{m+1,m}/v_{m+1}^T c$ and let $f \in \mathbb{R}^m$ such that

$$\begin{aligned} \beta_m c &= V_m f + \beta_m (v_{m+1}^T c) v_{m+1} \\ &= V_m f + h_{m+1,m} v_{m+1}, \end{aligned} \tag{2.3.3}$$

according to (2.3.3) and (2.2.4) we have:

$$\begin{aligned} AV_m - V_m(H_m - f e_m^T) &= h_{m+1,m} v_{m+1} e_m^T + V_m f e_m^T, \\ &= h_{m+1,m} v_{m+1} e_m^T + (\beta_m c - h_{m+1,m} v_{m+1}) e_m^T. \end{aligned}$$

Which shows (2.3.2) □

Note that $H_m - f e_m^T$ is upper Hessenberg.

Thus the lemma 2.3.1 shows that if we choose v_1 so that $c \in K_{m+1}(A, v_1)$ then l 'equality (2.3.2) is of the same form as (2.2.3) except for a multiplicative constant.

The following results show how to find such a Krylov space and how to obtain the relation (2.2.7)

Lemma 2.3.2. *[6] Let V_m and H_m be the matrices constructed by the Arnoldi process and let p be a polynomial of degree less than m then*

$$p(A)v_1 = V_m p(H_m)e_1.$$

proof. ,

it suffices to check that: $A^j v_1 = V_m H_m^j e_1 \quad 0 \leq j < m$ and the lemma goes away deducted. □

Lemma 2.3.3. *[6] Let $H_{m+1} \in M_{m+1}(\mathbb{R})$ be an upper Hessenberg matrix and p a monic polynomial of degree m (the coefficient of highest degree is equal to 1). SO*

$$e_{m+1}^T p(H_{m+1})e_1 = \prod_{j=1}^{m} h_{j+1,j}.$$

Theorem 4. *[6] Let c be the vector defined by (2.2.1). We assume that*

$\sigma(A) \cap \mathbb{P} = \emptyset$. *We define the monic polynomial*

$$p_m(t) = \prod_{j=1}^{m}(t - \mu_j) \tag{2.3.4}$$

and x the solution unique of

$$p_m(A)x = c \tag{2.3.5}$$

and let V_m, H_m, v_{m+1} and $h_{m+1,m}$ be determined by the Arnoldi process with $v_1 = \frac{x}{\|x\|}$. Let $\beta_m = h_{m+1,m}/v_{m+1}^T c$ and $f = \beta_m V_m^T c$, then

$$c \in K_{m+1}(A, v_1), \tag{2.3.6}$$

and

$$\sigma(H_m - f e_m^T) = \mathbb{P}. \tag{2.3.7}$$

proof. ,

The relation (2.3.6) comes from (2.3.5) and from (2.3.4) indeed

$$\begin{aligned} c = p_m(A)x \quad &\Longrightarrow \quad c = \underbrace{\|x\| p_m(A)}\, v_1 \qquad (\text{ because } v_1 = \frac{x}{\|x\|}) \\ &\Longrightarrow \quad c = g_m(A)v_1 \qquad (\text{with } g_m(t) = \|x\| p_m(t)) \\ &\Longrightarrow \quad c \in K_{m+1}(A, v_1) \qquad (\text{by definition of Krylov subspace}) \end{aligned}$$

To show the relation (2.3.7)

$$\begin{aligned} f &= \beta_m V_m^T c \\ &= \beta_m V_m^T p_m(A)x \\ &= \beta_m \|x\| V_m^T p_m(A)v_1, \end{aligned} \tag{2.3.8}$$

we set:$p_{m-1}(t) = \prod_{j=1}^{m-1}(t - \mu_j)$ thus replacing $p_m(t) = (t - \mu_m)p_{m-1}(t)$ on the right in the expression (2.3.8) and applying the lemma 2.3.2 we have

$$\begin{aligned} \beta_m \|x\| V_m^T p_m(A)v_1 &= \beta_m \|x\| V_m^T (A - \mu_m I)p_{m-1}(A)v_1 \\ &= \beta_m \|x\| V_m^T (A - \mu_m I)V_m p_{m-1}(H_m)e_1 \\ &= \beta_m \|x\| (H_m - \mu_m I)p_{m-1}(H_m)e_1 \\ &= \beta_m \|x\| p_m(H_m)e_1 \\ &= \beta_m \|x\| s, \end{aligned}$$

where s is defined by the 3 theorem.

In addition we have

$$\begin{aligned}\beta_m\|x\| &= \frac{h_{m+1,m}}{v_{m+1}^T c}\|x\| \\ &= \frac{h_{m+1,m}}{v_{m+1}^T p_m(A)x}\|x\| \\ &= \frac{h_{m+1,m}}{v_{m+1}^T p_m(A)v_1},\end{aligned}$$

so, by lemma 2.3.2 and lemma 2.3.3, we have

$$\begin{aligned}\frac{h_{m+1,m}}{v_{m+1}^T p_m(A)v_1} &= \frac{h_{m+1,m}}{v_{m+1}^T V_{m+1} p_m(H_{m+1})e_1} \\ &= \frac{h_{m+1,m}}{\prod_{j=1}^{m} h_{j+1,j}} \\ &= \frac{1}{\prod_{j=1}^{m-1} h_{j+1,j}}\end{aligned}$$

Thus we deduce that $\alpha = \beta_m\|x\|$ (because $\alpha = \prod_{j=1}^{m-1} h_{j+1,j}^{-1}$) and it follows that $\alpha s = f$ and by theorem 4 we have

$$\sigma(H_m - fe_m^T) = \mathbb{P}.$$

□

Remark 5. *In practice, we do not use the scalar β_m as it is defined in theorem 4 because the numerical results obtained will be worse than those obtained by this expression*

$$\beta_m = \frac{c^T d}{\|c\|^2}. \tag{2.3.9}$$

with $d = h_{m+1,m}v_{m+1} + V_m f$.

We have by the Arnoldi process

$$AV_m - V_m H_m = h_{m+1,m} v_{m+1} e_m^T,$$

adding $V_m f e_m^T$ to both members of the equality we have

$$AV_m - V_m H_m + V_m f e_m^T = h_{m+1,m} v_{m+1} e_m^T + V_m f e_m^T \tag{2.3.10}$$

so

$$AV_m - V_m(H_m - f e_m^T) = (h_{m+1,m} v_{m+1} + V_m f) e_m^T$$

we put

$$H = H_m - f e_m^T,\ d = h_{m+1,m} v_{m+1} + V_m f$$

and we will not fail to point out that d is the last column of the matrix $AV_m - V_m H$ which means that we have

$$AV_m - V_m H = d e_m^T$$

and to get back to the relation (2.2.3) we must that d is proportional to c

$$d = \beta_m c$$

so we get

$$\begin{aligned} d = \beta_m c &\Rightarrow c^T d = \beta_m c^T c \\ &\Rightarrow \beta_m = \frac{c^T d}{c^T c} \\ &\Rightarrow \beta_m = \frac{c^T d}{\|c\|^2}. \end{aligned}$$

we have

$$AV_m - V_m H = \beta_m c e_m^T$$

and therefore by identification with the relation (2.2.3), the solution X sought is given by

$$X = \frac{1}{\beta_m} V_m.$$

Now that we know how to obtain the matrices H and X, we can present the algorithm for partial placement of the poles of the matrix H_m [3]:

Algorithm 4. *Placement of the poles of* H_m

$l_1 = e_1$;

$\alpha = 1$;

for $i = 1, ..., m-1$ *do*

$l_{i+1} = (H_m - \mu_i I) l_i$;

$\alpha = \alpha . h_{i+1,i}^{-1}$;

End i

$s = (H_m - \mu_m I) l_m$;

$f = \alpha s$;

calculate $H_m - f e_m^T$;

End

Now it only remains to know how to obtain the vector v_1.

Conclusion

Solving Sylvester's equation by Arnoldi's process therefore leads us to solve two problems.

The first lies in determining the vector v_1 to start Arnoldi's process, and the second problem is a placement of poles on the matrix H_m.

We come back to the first problem in more detail in Chapter 3.

Chapter 3

Saad and Datta method

In this chapter we solve the system $q_m(A)x = c$, which constitutes the first step of the Saad-Datta method [3] (see algorithm 6, page 46) for solving Sylvester's equation. The importance of this step lies in the fact that it makes it possible to find the vector v_1, which will start the process of Arnoldi.

Throughout this chapter, $\mathbb{P} = \{\mu_1, ..., \mu_m\}$ is a set of m complex numbers with negative real part and different two two, $\sigma(A)$ means the spectrum of A and $q_m(t) = \prod_{i=1}^{m}(t - \mu_i I)$.

3.1 Presentation of the problem

We know that the vector v_1 must satisfy the relation (2.2.6) (chapter 2 page 34) which leads to the fact that v_1 is a solution of the following equation

$$q_m(A)x = c, \tag{3.1.1}$$

knowing that

$$q_m(t) = (t - \mu_1)...(t - \mu_m),$$

that's to say

$$(A - \mu_1 I)...(A - \mu_m I)x = c, \tag{3.1.2}$$

by setting $x = x_m$ and $x_0 = c$, the relation (3.1.2) becomes

$$(A-\mu_1 I)...(A-\mu_m I)x_m = x_0, \tag{3.1.3}$$

we put $(A-\mu_m I)x_m = x_{m-1}$ so the relation (3.1.3) becomes

$$(A-\mu_1 I)...(A-\mu_{m-1} I)x_{m-1} = x_0,$$

repeating the same operation, we get

$$\begin{cases} (A-\mu_i I)x_i = x_{i-1} \\ i = 1, ..., m \end{cases} \tag{3.1.4}$$

Thus, solving (3.1.1) is equivalent to solving (3.1.4). The last vector x_m is of course the solution wanted. However, this method cannot be used for two reasons. The first is that it is very expensive; because the methods direct are not possible in the case of large linear systems. The second reason is due to possible disruptions in iterations; see [1] for details.

3.2 Rational fraction approach

If the resolution of the m linear systems cannot succeed for the reasons already cited, then a different approach is called for.

We know from (3.1.1) that

$$x = [q_m(A)]^{-1}c, \tag{3.2.1}$$

and we also have

$$\frac{1}{q_m(t)} = \frac{1}{\prod_{i=1}^{m}(t-\mu_i)}, \tag{3.2.2}$$

whether we denote by $q'_m(\mu_i)$ the derivative of $q_m(t)$ calculated at point μ_i we have

$$q'_m(\mu_i) = \prod_{\substack{j=1 \\ j\neq i}}^{m} (\mu_i - \mu_j),$$

the derived numbers $q'_m(\mu_i)$ are all different from zero, this is due to the fact that all the elements of $\mathbb{P}$ are disjoint

two by two, thus the decomposition into simple elements of (3.2.2) gives

$$\frac{1}{q_m(t)} = \sum_{i=1}^{m} \frac{1}{q'_m(\mu_i)(t-\mu_i)}, \tag{3.2.3}$$

so the relation (3.2.1) becomes

$$x = \sum_{i=1}^{m} \frac{1}{q'_m(\mu_i)} (A - \ mu_i I)^{-1} c, \tag{3.2.4}$$

the expression (3.2.4) makes no sense if either matrices $(A - \mu_i I)$ is not invertible. Since $\mathbb{P} \cap \sigma(A) = \emptyset$, the spectrum of the matrix A does not contain any root of the polynomial $q_m(t)$ and therefore the matrices $(A - \mu_i I)$ are all invertible.

Indeed if $\mu \notin \sigma(A)$ we have

$$\begin{aligned} \mu \notin \sigma(A) &\Leftrightarrow \mu \text{ is not the root of the characteristic polynomial of } A \\ &\Leftrightarrow det(A - \mu I) \neq 0 \\ &\Leftrightarrow (A - \mu I) \text{ invertible,} \end{aligned}$$

which proves what we just said.

We pose

$$x_i = (A - \mu_i I)^{-1} c,$$

therefore the relation (3.2.4) becomes

$$x = \sum_{i=1}^{m} \frac{1}{q'_m(\mu_i)} x_i, \tag{3.2.5}$$

so to find x we have to solve the m linear systems next

$$(A - \mu_i I) x_i = c \ \text{ with } i = 1, ..., m, \tag{3.2.6}$$

and then use the linear combination given by (3.2.5) to find x.

For the resolution of (3.2.6) we use the GMERS method [12]; however the vector v_1 sought must be normalized, which is not necessarily the case for x, so we set

$$v_1 = \frac{x}{\|x\|}.$$

3.3 Resolution of $q_m(A)x = c$ by the GMRES method

The GMRES method [12] is one of the most efficient methods for solving type systems (3.2.6). To apply the GMRES method, we construct V_l and H_l by the Arnoldi process applied to A and $v_1 = \frac{c}{\|c\|}$, this base will be used in all m systems given by (3.2.6) because Arnoldi's base , given by the matrix V_l, does not vary if we replace A by $(A - \sigma I)$ (where σ is a scalar). Arnoldi's process gives

$$AV_l = V_l H_l + h_{l+1,l} v_{l+1} e_l^T,$$

if we add $-\sigma V_l$ to both members of the previous equality

$$AV_l - \sigma V_l = V_l H_l + h_{l+1,l} v_{l+1} e_l^T - \sigma V_l,$$

we then obtain

$$(A - \sigma I)V_l = V_l(H_l - \sigma I) + h_{l+1,l} v_{l+1} e_l^T. \tag{3.3.1}$$

The interest of this idea lies in the fact that the same basis V_l will be used to find all vectors x_i which makes it possible to save in the calculations because we just modify the matrix H_l by replacing it with $H_l - \mu_i I$ for $i = 1, ..., m$.

Remark 6. - *We choose the integer l as being the smallest integer such that*

$$\|r_i^{(l)}\| < \epsilon \text{ for } i = 1, ..., m \tag{3.3.2}$$

with $r_i^{(l)} = c - (A - \mu_i I)x_i$ for $i = 1, ..., m$ (see [6] for more details).

Finally we solve the system given by (3.1.1) by the following Algorithm

Algorithm 5. *Solving* $q_m(A)x = c$

1. *Compute* $\beta = \|c\|$ *and* $v_1 = \frac{c}{\|c\|}$;

2. *Apply the Arnoldi process on the matrix* A *and* v_1 *to construct the matrices* H_l *and* V_l;

3. *For* $i = 1, ..., m$ *do*

   calculate $H_l^{(\mu_i)} = H_l - \mu_i I$; *find* $y_l^{(\mu_i)} = argmin_{y \in \mathbb{C}^l} \|\beta e_1 - \tilde{H}_l^{(\mu_j)} y\|$ *where* $\tilde{H}_l^{(\mu_j)} = \begin{pmatrix} H_l^{(\mu_i)} \\ 0 \dots h_{l+1,l} \end{pmatrix}$;

   if $\|c - \tilde{H}_l^{(\mu_j)} y_l^{(\mu_i)}\| \geq$ *epsilon*

   choose larger l *and go to 2.*

   end if

   calculate $x_i = V_l y_l^{(\mu_i)}$;

   end i

4. *calculate* $x = \sum_{i=1}^{m} \frac{1}{q'_m(\mu_i)} x_i$

3.4 The choice of poles

The eigenvalues of A are not known beforehand, so it is difficult to choose in advance the set of poles $\mathbb{P}$ of way that $\mathbb{P} \cap \sigma(A) = \emptyset$. If one of the μ_i is close to one of the eigenvalues of A, the linear system which associated with it $(A - \mu_i I)x_i = c$ will be ill-conditioned. To solve this problem one can make an estimate on the eigenvalues of A, such an estimate makes it possible to choose the elements of $\mathbb{P}$ well. Other methods allow the estimation of the eigenvalues of A, like the deflation method, combined with polynomial iterations, as in [3], or Arnoldi's method restarted implicitly (IRA method: Implicitly restarted Arnoldi) which gives an estimate of the eigenvalues of the matrix A, as in the works of L.Reichel, B.Lewis and D.Calvetti (for more details see [6]). It should be noted that in several situations in practice, such as the case of large and flexible spatial structures (LFSS) [3],

the eigenvalues of the matrix A are given by an analytical formula (see page 49) which facilitates the choice of the elements of $\mathbb{P}$.

3.5 Algorithm for Solving Sylvester's Equation

Now we know how to get x, and at the same time the vector $v_1 = \frac{x}{\|x\|}$ which allows to have $v_{m+1} = \alpha c$, and since we have already seen in chapter 2 how to calculate the matrices H and X, we can then give the definitive algorithm for the resolution of Sylvester's equation.

Algorithm 6. *Solving Sylvester's equation*

1. *solve $q_m(A)x = c$ by the GMRES method (i.e. the algorithm 5), then calculate $v_1 = \frac{x}{\|x\|}$*
2. *run Arnoldi's process to calculate V_m and H_m*
3. *find y such that $\sigma(H) = \sigma(H_m - ye_m^T) = \mathbb{P}$*
4. *calculate $\beta_m = c^T d/\|c\|^2$ with d being the last column of $AV_m - V_m H$*
5. *$X_m = \frac{1}{\beta_m} V_m$*

Conclusion

Since the matrices A and $A - \mu_i I$ have the same Krylov subspace, this makes it possible to make significant savings in calculation and in memory, because only the step of solving the minimization problem (in the method of GMRES) which will be executed as many times as the number elements of $\mathbb{P}$. Regarding the choice of the set $\mathbb{P}$, it is essential to choose it well because otherwise the system (3.1.1) will be badly conditioned.

Chapter 4

Numerical tests[5]

In this chapter, we give numerical tests for details, we can consult the article of C.W Gear [5]. The tests were done we have PC equipped with an Intel processor. Celeron D at 3.2 GHz, 2 GB of RAM, precision $2,2204.10^{-16}$ and equipped with version 7.4 of Matlab.

In all the tests, the vector c is generated randomly by the command **rand** of matlab (taking into account the dimension of the matrix A). The vector v_1, solution of (3.2.6), is determined by the algorithm 5 (page 45) with $\epsilon = 10^{-8}$. It should also be noted that before starting we make sure that the spectrum of A and the set $\mathbb{P}$ are disjoint.

In this chapter, we denote by $\|u\|$ the Euclidean norm [1] of the vector $u \begin{pmatrix} u_1 \\ \vdots \\ u_k \end{pmatrix}$ given by

$$\|u\| = \sqrt{\sum_{i=1}^{k} |u_i|^2}$$

and by $\|U \mid$ the spectral norm [1] of the matrix $U = (u_{ij})_{1<i,j<k}$ given by

$$\|U\| = \sqrt{\rho}$$

with ρ is the largest eigenvalue of the matrix $A^T A$

4.1 Gear Matrix

In this test we apply the algorithm 6 (page 46) by choosing $l = 16$ on the matrix A of size $n = 1000$ given by

$$A = \begin{pmatrix} 1 & 1 & 0 & \dots & \dots & 0 \\ 1 & 0 & 1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 1 \\ 0 & \dots & \dots & 0 & 1 & 0 \end{pmatrix} \tag{4.1.1}$$

this type of matrix is called **Gear matrix** ("**Gear matrix**"), and for more details on this type of matrix we return readers to the C.W Gear article [5]. We chose the next set of poles

$$\mathbb{P} = \{\mu_k = -4k/k = 1, ..., m\}. \tag{4.1.2}$$

After running the 6 algorithm (page 6) we calculate the norms

$\|(AX - XH) - ce_m^T\|$ and $\|\sigma(H_m - fe_m^T) - \mu\|$ where $\mu \in \mathbb{P}$, $H = H_m - fe_m^T$; the results obtained are reported in the following table

Remark 7. *If $l < 16$, the algorithm 6 stops; but the precision increases by taking l larger, as shown in the table 4.2*

m	$\|(AX - XH) - ce_m^T\|$	$\|\mu - \sigma(H_m - fe_m^T)\|$
4	$3,3557.10^{-8}$	$1,3902.10^{-13}$
6	$5,3488.10^{-8}$	$2,8664.10^{-11}$
8	$7,5778.10^{-8}$	$2,1207.10^{-10}$
10	$1,0016.10^{-7}$	$3,4674.10^{-8}$
12	$1,2644.10^{-7}$	$6,2719.10^{-7}$
14	$1,5445.10^{-7}$	$2,4907.10^{-4}$

Table 4.1: Results obtained for Gear matrix A of size $n = 1000$ and $l = 16$

Remark 8. *Note that the greater the number of poles, the greater the norms $\|(AX - XH) - ce_m^T\|$ and $\|\sigma(H_m - fe_m^T) - \mu\|$ increase.*

We redo the same work done previously, except that here the matrix A is a Gear matrix of size $n = 500$, and we keep the same set $\mathbb{P}$ used in test 1

m	$\|(AX - XH) - ce_m^T\|$	$\|\mu - \sigma(H_m - fe_m^T)\|$
4	$3.9215.10^{-14}$	$2.8891.10^{-13}$
6	$1,6637.10^{-13}$	$4,4564.10^{-11}$
8	$7,3598.10^{-13}$	$6,5515.10^{-10}$
10	$2,9257.10^{-12}$	$1,4474.10^{-8}$
12	$1,0948.10^{-11}$	$3,4224.10^{-6}$
14	$4,2106.10^{-11}$	$4,8322.10^{-5}$

Table 4.2: Results obtained matrix A of Gear of size $n = 1000$ and $l = 32$

m	$\|(AX - XH) - ce_m^T\|$	$\|\mu - \sigma(H_m - fe_m^T)\|$
4	$2,3109.10^{-8}$	$4,6729.10^{-14}$
6	$3,6771.10^{-8}$	$1,7709.10^{-11}$
8	$5,2031.10^{-8}$	$1,3948.10^{-9}$
10	$6,8712.10^{-8}$	$1,0157.10^{-8}$
12	$8,6675.10^{-8}$	$1,5981.10^{-6}$
14	$1,0581.10^{-7}$	$1,5231.10^{-4}$

Table 4.3: Results obtained matrix A of Gear of size $n = 500$ and $l = 16$

4.2 LFSS Matrix

We consider the matrix A defined by

$$A = \begin{pmatrix} 0_p & I_p \\ L & D \end{pmatrix} \tag{4.2.1}$$

where $n = 2p$ is the size of this matrix ($p = 500$ for this test), and the matrices D, L are defined as follows

$D = diag\{d_1, ..., d_p\}$ and $L = diag\{l_1, ..., l_p\}$, the eigenvalues of this matrix are the solutions of the equations

$$x^2 - d_k x - l_k = 0, \qquad k = 1, ..., p \tag{4.2.2}$$

Thus if $d_k = 2\alpha_k$ and $l_k = -(\alpha_k^2 + \beta_k^2)$ the spectrum of A is given by

$\sigma(A) = \{ lambda_k, \bar{\lambda_k}\}$ with $\lambda_k = \alpha_k + i\beta_k$. This type of matrix is used to model large and flexible spatial structures (LFSS for Large flexible space structures) [3].

To illustrate the impact of the choice of the set of poles on the results obtained, we choose 2 sets of poles that we define by

m	$\|(AX - XH) - ce_m^T\|$	$\|\mu - \sigma(H_m - fe_m^T)\|$
4	$3,1980.10^{-14}$	$1,5291.10^{-13}$
6	$1,4500.10^{-13}$	$6,9802.10^{-11}$
8	$6,1146.10^{-13}$	$1,4631.10^{-9}$
10	$2,3719.10^{-12}$	$2,3719.10^{-8}$
12	$9,2488.10^{-12}$	$4,3654.10^{-6}$
14	$3,4298.10^{-11}$	$1,2756.10^{-4}$

Table 4.4: Results obtained matrix A of Gear of size $n = 500$ and $l = 32$

$$\mathbb{P}_1 = \{\mu_k = a_k + ib_k/a_k \in [-2, -1] \text{ and } b_k \in [0, 1]\} \tag{4.2.3}$$

and

$$\mathbb{P}_2 = \{\mu_k = a_k + ib_k/a_k \in [-4, -3] \text{ and } b_k \in [0, 1]\} \tag{4.2.4}$$

In both cases, we give the minimum value l so that the algorithm does not stop.

The following table illustrates the results obtained for the set $\mathbb{P}_1$,

l	m	$\|(AX - XH) - ce_m^T\|$	$\|\mu - \sigma(H_m - fe_m^T)\|$
105	4	$3,1328.10^{-8}$	$1,6338.10^{-14}$
151	6	$6,8162.10^{-7}$	$2,6208.10^{-13}$
190	8	$5,5369.10^{-6}$	$1,6679.10^{-12}$

Table 4.5: Results obtained for the set of poles $\mathbb{P}_1$

and this one illustrates the results obtained for the set $\mathbb{P}_2$,

l	m	$\|(AX - XH) - ce_m^T\|$	$\|\mu - \sigma(H_m - fe_m^T)\|$
21	4	$1,1413.10^{-6}$	$6,8883.10^{-13}$
21	6	$1,2864.10^{-4}$	$2,8424.10^{-11}$
21	8	$0,0084$	$1,6200.10^{-8}$

Table 4.6: Results obtained for the set of poles $\mathbb{P}_2$

The tables 4.5 and 4.6 show that the set $\mathbb{P}_1$ gives better results than the set $\mathbb{P}_2$.

4.3 Graphical representation of spectra

In this example we graphically represent the distribution of the spectra of the matrices A, $H = H_m - fe_m^T$, and H_m in the plane complex.

We consider the Saad matrix $A = A(r, s)$, of size $n = 500$ (with $s = 10$, $r = 50$ and $n = rs$), defined as follows

$$\begin{cases} a_{i,i} = 4 & \text{for } i = 1, ..., n \\ a_{i,i+1} = -1.2 & \text{for } i = 1, ..., n-1 \\ a_{i,i-1} = 1.1 & \text{for } i = 2, ..., n \\ a_{ir+1,ir} = a_{ir,ir+1} = 0 & \text{for } i = 1, ..., s-1 \\ a_{i+r,i} = a_{i,i+r} = -1 & \text{for } i = 1, ..., (s-1)r \end{cases} \tag{4.3.1}$$

the set of poles chosen for this test is the set defined by (4.2.3) and we take $m = 6$.

conclusion

In this work, we presented an algorithm based on the Arnoldi process for the resolution of the Sylvester equation, in the case of sparse and wide matrices, while assuming that the second member is a matrix of rank 1.

The fact that the second member of Sylvester's equation has rank 1 made it possible to make the analogy with the Arnoldi process. thing that has gave the idea to use this process to find the solution.

The upper Hessenbergue matrix given by the Arnoldi process is not always stable and does not necessarily satisfy the condition of existence and uniqueness of the solution, so we made a placement of poles on it to remedy this problem.

It has also been pointed out that the choice of the set of poles can distort the solution if it is not done adequately because if a pole is close to an eigenvalue of the matrix A the system which is associated is badly conditioned, which falsifies the results by the following.

One can exploit the fact that the spectrum, in several situations in practice, is given by an analytical formula to choose a set adequate poles. In the absence of such a formula, the use of the estimates of the largest and the smallest eigenvalue of the matrix in question is always possible.

The generalization of this work to the case where the matrix C has any rank r remains possible. We follow the same pattern, except that in this case we base ourselves on the Arnoldi process by blocks and the GMRES method by blocks.

Bibliography

[1] B.N. Datta, *Numerical methode for linear control systems design and analysis, Departement of mathematical Sciences Northern Illinois University Dekalb, IL 60115 (2003).*

[2] B.N. Datta, *An algorithme to assigne eigeinvalues in a Hessenberg matrix, IEEE Trans. Autom. Control, AC-32 (1987), pp. 414-417.*

[3] B.N. Datta and Y. Saad, *Arnoldi Methods for large Sylvester-Like Observer Matrix Equation, and an Associated for Partial Spectrum Assignement, Linear Algebra and Its Applications 154-156:225-244 (1991).*

[4] L. Elbouyahyaoui, *Etude du polynôme minimal pour la méthode du GMRES cas standard et cas par blocs, Memoire de DESA, EMI (2005).*

[5] C.W. Gear, *A simple set of test matrices for eigenvalue programs, Math. Comp., 23:119-125, (1969).*

[6] B. Lewis, D. Calvetti, L. Reichel, *On the solution of large Sylvester observer equation, Numer. Linear Algebra Appl. (2001); 8:1-16.*

[7] B. Lewis, D. Calvetti, L. Reichel, *Partial eigenvalue assignement for large linar control systems, Contemporary Mathematics. Primary 93B55, 65F15 (1991).*

[8] B. Lewis, D. Calevetti, L. Reichel, *On the selection of pole placement problem, linéar algebra Appl., 302-303 (1999), pp.331-345.*

[9] A. Messaoudi, *Recursive interpolation algorithm: A formalism for solving systems of equations-II iterative methodes, Journal of computational and applied Mathematics, 76 (1996) 31-53.*

[10] A. Rachid et D. Mehdi, *Réduction,réalisation et commande des systèmes linéaires, edition: Scientifika* 1^{er} *trimestre 1993, ISBN: 2-909894-03-7.*

[11] Y. Saad, *Projection and deflation methods for partial pole assignement in linear stat feedback, IEEE Trans. Autom. Control, AC-33 (1988), pp. 290-297.*

[12] Y. Saad et M.H. Schultz, *GMRES a generalized minimal residual algoritme for solving nonsymetrique linéar equation, SIAM J. Sci. Statist Comput 7 (1986) 856-869.*

[13] D.C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi methode, SIAM J.Matrix Anal. Appl., 13 (1992), pp.357-385.*

[14] W.M. Wonham, *Linear Multivariate Control: A Geometric Approach, 3rd ed., Springer, New York, (1985).*